



Escuela  
Politécnica  
Superior

# Desarrollo de arquitecturas especializadas para Sistemas de Conducción Inteligente



Grado en Ingeniería Informática

## Trabajo Fin de Grado

Autor:

D. Daniel Ortega Fernández

Tutor/es:

D. Antonio Jimeno Morenilla

D. Higinio Mora Mora



Universitat d'Alacant  
Universidad de Alicante

Septiembre 2017

# Índice

Índice de figuras .....	4
Introducción .....	6
Estado del arte .....	8
OBD-II, tratamiento y obtención de datos .....	8
Importancia del estado del conductor .....	10
La nube, procesamiento de datos a gran escala .....	11
Definición redes neuronales artificiales, mapa autoorganizado (SOM) .....	15
Algoritmo SOM .....	17
Interpretación SOM .....	18
Trabajos previos relacionados sobre mapas autoorganizados .....	19
Objetivos .....	21
Cuerpo del trabajo .....	22
Diseño de la arquitectura .....	22
Parte 1. Vector de datos .....	22
Objetivos y método .....	22
Obtención de datos .....	23
Datos utilizables .....	23
Requisitos funcionales .....	25
Parte 2. Almacenaje de datos .....	26
Objetivos y método .....	26
Características del almacenaje de datos .....	27
Requisitos funcionales .....	27
Requisitos no funcionales .....	27
Parte 3. Tratamiento de datos .....	28
Objetivos y método .....	28
Red neuronal utilizada .....	29
Problemas mapa autoorganizado .....	29
Ventajas mapa autoorganizado .....	30
Requisitos funcionales .....	30
Requisitos no funcionales .....	31
Parte 4. Clasificación de datos .....	31
Objetivos y método .....	31

Criterio de clasificación.....	32
Requisitos funcionales.....	34
Implementación del prototipo.....	34
Parte 1. Recogida y preparación de datos .....	34
Elaboración de circuitos .....	34
Circuitos de conducción rápida .....	35
Circuitos de callejeo .....	36
Circuitos mixtos.....	38
Vehículo de pruebas .....	39
Parte 2. Red neuronal, mapa autoorganizativo .....	40
Análisis de datos recogidos .....	40
Carga del motor .....	40
Temperatura del refrigerante del motor .....	42
RPM del motor .....	43
Velocidad del vehículo.....	45
Temperatura en la toma de aire.....	46
Posición del pedal del acelerador D .....	48
Implementación SOM .....	49
Configuración red SOM .....	50
Parte 3. Resultados y clasificación .....	51
Resultados SOM carga del motor .....	51
Resultados SOM temperatura del refrigerante del motor .....	52
Resultados SOM RPM del motor .....	54
Resultados SOM velocidad del vehículo .....	55
Resultados SOM temperatura en la toma de aire .....	56
Resultados SOM posición del pedal del acelerador D .....	58
Clasificación .....	59
Conclusiones.....	60
Bibliografía.....	61

# Índice de figuras

Imagen 1: Esquema diseño .....	7
Imagen 2: Arquitectura red aprendizaje competitivo [24].....	16
Imagen 3: Algoritmo mapa autoorganizado[23].....	17
Imagen 4: Ejemplo de sensores y OBD-II [33] .....	22
Imagen 5: Descripción de modos disponibles OBD-II [34] .....	24
Imagen 6: Descripción de valores útiles [34].....	25
Imagen 7: Ejemplo de organización de los datos en la BD .....	26
Imagen 8: Imagen ejemplo de mapa autoorganizado [39].....	28
Imagen 9: Tabla ejemplo Parámetros/Características.....	33
Imagen 10: Circuito conducción rápida 1 .....	35
Imagen 11: Circuito conducción rápida 2 .....	36
Imagen 12: Circuito callejeo 1 .....	36
Imagen 13: Circuito callejeo 2.....	37
Imagen 14: Circuito callejeo 3.....	37
Imagen 15: Circuito mixto 1 .....	38
Imagen 16: Circuito mixto 2 .....	39
Imagen 17: Tabla comparativa carga del motor conducción rápida vs callejear .....	41
Imagen 18: Tabla comparativa carga del motor conducción mixta .....	41
Imagen 19: Tabla comparativa temperatura del refrigerante del motor conducción rápida vs callejear .....	42
Imagen 20: Tabla comparativa temperatura del refrigerante del motor conducción mixta .....	43
Imagen 21: Tabla comparativa RPM del motor conducción rápida vs callejear .....	44
Imagen 22: Tabla comparativa RPM del motor conducción mixta .....	44
Imagen 23: Tabla comparativa velocidad del vehículo conducción rápida vs callejear	45
Imagen 24: Tabla comparativa velocidad del vehículo conducción mixta.....	46
Imagen 25: Tabla comparativa temperatura en la toma de aire conducción rápida vs callejear .....	47
Imagen 26: Tabla comparativa temperatura en la toma de aire conducción mixta .....	47
Imagen 27: Tabla comparativa posición del pedal del acelerador D conducción rápida vs callejear.....	48
Imagen 28: Tabla comparativa posición del pedal del acelerador D conducción mixta	49
Imagen 29: SOM carga del motor, impactos .....	51
Imagen 30: SOM carga del motor, distancias .....	52

Imagen 31: SOM temperatura refrigerante del motor, impactos.....	53
Imagen 32: SOM temperatura refrigerante del motor, distancias .....	53
Imagen 33: SOM RPM del motor, impactos .....	54
Imagen 34: SOM RPM del motor, distancias .....	54
Imagen 35: SOM velocidad del vehículo, impactos .....	55
Imagen 36: SOM velocidad del vehículo, distancias .....	56
Imagen 37: SOM temperatura toma de aire, impactos .....	57
Imagen 38: SOM temperatura toma de aire, distancias .....	57
Imagen 39: SOM posición pedal acelerador D, impactos .....	58
Imagen 40: SOM posición pedal acelerador D, distancias .....	58
Imagen 41: Clasificación de parámetros final.....	59

# Introducción

Este trabajo proporciona un diseño que busca dotar de utilidad la gran cantidad de información que pueden proporcionar los vehículos en su conjunto. Se busca analizar datos obtenidos de la conducción para determinar distintos parámetros sobre la misma o sobre el conductor.

El análisis de la conducción y sus posibilidades entran en el ámbito de la “*smart city*” y la búsqueda de dotar de inteligencia a elementos que participan de forma activa en la vida cotidiana de las personas.

Por otro lado, se utilizarán técnicas como las redes neuronales y el “*big data*” para lograr clasificar la cantidad de información que generan los vehículos, analizarlos y encontrar patrones.

Todo esto con el objetivo de determinar si el comportamiento del conductor y el uso que este da al vehículo es óptimo o no. Además de obtener la capacidad de clasificar datos de otros conductores y determinar el comportamiento de un número mayor de muestras.

El objetivo es desarrollar un modelo de análisis de datos que permita a cualquier empresa relacionada con los vehículos, como, por ejemplo, empresas de taxis, transporte de mercancías, conductores particulares o empresas de transporte público mediante autobuses, averiguar el uso que los conductores hacen de los vehículos. De esta forma, las empresas o particulares podrían ahorrar costes derivados de la mala conducción como el aumento del consumo de la gasolina.

Con el fin de lograr dicho objetivo se modela el siguiente sistema:

En primer lugar, partimos de datos de vehículos tales como revoluciones por minuto del motor, velocidad del automóvil o porcentaje de aceleración. Dispondremos de estos datos y los prepararemos para el siguiente paso.

Una vez tenemos los vectores de datos organizados utilizaremos una red neuronal artificial denominada mapa auto-organizado que nos permita clasificar dichos datos.

Finalmente, generaremos una tabla con las diferentes características que podamos extraer de los datos y la clasificación obtenida. Estas características corresponderán a diferentes aptitudes del conductor frente a la conducción, el grado de despiste, por ejemplo. Estas aptitudes, serán organizadas en grados que irán desde una conducción óptima a una conducción agresiva.

El reto, por lo tanto, es lograr una clasificación efectiva de los datos y que podamos determinar si la conducción que se está llevando a cabo podría mejorarse.

A continuación, podemos ver una representación del sistema:

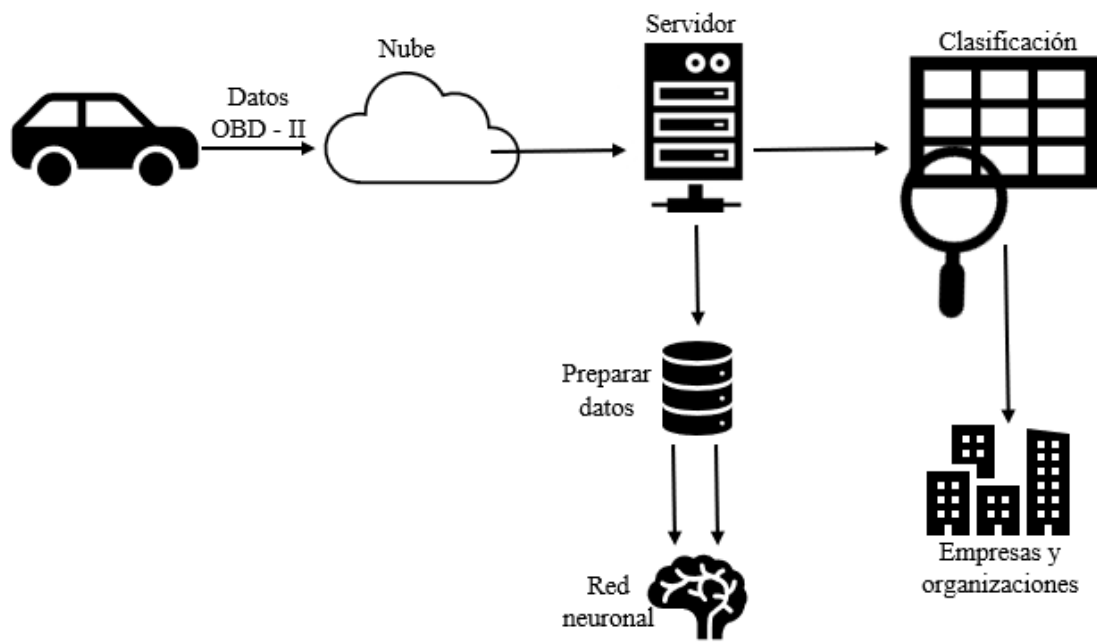


Imagen 1: Esquema diseño

# Estado del arte

## OBD-II, tratamiento y obtención de datos

Los datos ya recogidos que trataremos en este trabajo vienen dados a partir de OBD-II. Son numerosos los trabajos que tratan el tema del tratamiento de datos OBD-II recogidos de vehículos, la utilidad que se puede sacar a estos datos y las técnicas que se utilizan para lograr este fin.

En 2015 mediante el trabajo “*Cloud-based driver monitoring and vehicle diagnostic with OBD2 telematics*” [1] los investigadores M. Amarasinghe, S. Kottegoda, A. L. Arachchi, S. Muramudalige, H. M. N. D. Bandara y A. Azeez desarrollan una arquitectura en la nube que permite monitorizar un vehículo y realizar un diagnóstico del mismo a través de los datos recogidos en OBD-II. El sistema es dividido en tres partes, el lector OBD-II, una aplicación móvil y un backend cloud que recoge los datos y realiza diagnósticos. El sistema posibilita reconocimiento de patrones de conducción mediante “*fuzzy logic*”. Por otro lado, los datos más utilizados son los recogidos por los sensores de MAF, oxígeno, temperatura de aceite y temperatura del motor. En este trabajo se concluye que existe camino en el análisis y la recolección de los datos mediante OBD-II.

Wonang Jang, Daeseong Jong y Dohoon Lee en 2016 mediante el trabajo “*Methodology to improve driving habits by optimizing the in-vehicle data extracted from OBDII using genetic algorithm*” [2] hablan de cómo mejorar los hábitos de un conductor mediante los datos recogidos de OBD-II y usando el “*Genetic algorithm*”. Se determina una solución óptima mediante simulación de conducción en el mismo tramo realizado por una población de cromosomas. La arquitectura del sistema se basa en obtener los datos de OBD-II y datos del GPS, codificarlos con los cromosomas, insertarlos dentro de una población de los mismos y buscar el patrón de mejor eficiencia de combustible. Este trabajo abre la posibilidad de que es posible conocer cuando se está realizando una conducción poco eficiente y saber cuál es el margen de mejora.

Mediante el artículo “*The Self-Induction System for Fuel-Efficient Driving using OBD II*” [3] en 2014 por el investigador W. Jang, se trata también el objetivo del ahorro del combustible, pero utilizando un sistema de autoinducción. Este método busca demostrar que es posible lograr el objetivo incluso con poco coste computacional, puesto que el dispositivo que recoge y trata los datos es el mismo smartphone.

Los investigadores M: J: Kim, J. W. Jang, y Y. S. Yu en 2010 con el artículo “*A study on in-vehicle system using OBD-II with navigation*” [4] estudian un Sistema que mediante la utilización de los sistemas de navegación y los datos recopilados en OBD-II sea posible descubrir cuando un vehículo va a tener un comportamiento anormal y este podría producir un accidente.



Para concluir, estas numerosas investigaciones nos proporcionan una visión de la importancia y las posibilidades que existen mediante el tratamiento de datos en OBD-II.

Estos trabajos, nos hacen ver que:

- Es posible determinar el comportamiento del conductor o la eficiencia del conductor.
- Un análisis a mayor escala con gran cantidad de datos y recogiendo históricos podría mejorar las predicciones, la calidad de la conducción y ahorrar costes.
- Los datos obtenidos a través de OBD-II son de gran utilidad y no ha sido explotado en su totalidad.
- Existe trabajo por hacer en cuanto a desarrollar una arquitectura que permita el aprovechamiento de estos datos.

# Importancia del estado del conductor

Conocer el estado del vehículo y del conductor es de una gran relevancia y existen diferentes investigaciones que tratan y evidencian como una parte de los accidentes es provocado por datos que podemos extraer y conocer del propio vehículo.

En el trabajo “*Driving anger, emotional and instrumental aggressiveness, and impulsiveness in the prediction of aggressive and transgressive driving*” [5] de los investigadores Berdoulat, E., Vavassori, D., y Sastre, M. en 2013, se pueden extraer diferentes conclusiones sobre como la agresividad afecta a la conducción o como el estado emocional del conductor es importante para la seguridad del mismo.

Mediante “*Aggressive driving: research update, American Automobile Assoc.*” [6] en 2009 se extraen conclusiones similares a las del documento anterior. La agresividad es un factor importante a la hora de conducir y este puede determinar que se produzca un accidente.

Mediante estos dos trabajos se pueden extraer conclusiones muy importantes:

- Como hemos visto en apartados anteriores la agresividad es un factor que es posible medir mediante mediciones realizadas en el vehículo.
- La agresividad y el estado emocional del conductor son factores que afectan a la conducción.
- La agresividad y el estado emocional pueden provocar accidentes de tráfico.

Por lo tanto, podemos concluir que la agresividad unido al estado del vehículo son factores de riesgo y que estos producen una parte de los accidentes en la actualidad. Conocer esto dota de mayor importancia cuando es posible obtener datos del vehículo que nos indican parámetros que permiten conocer el estado del conductor. Un análisis de estos datos podría reducir el número de accidentes o alertar al conductor de la situación para que este al menos conozca la situación de peligro en la que se encuentra.

# La nube, procesamiento de datos a gran escala

Una vez conocemos la importancia y las posibilidades que tienen los datos que podemos obtener de los vehículos conocemos las aportaciones que pueden tener el tratamiento de estos datos a gran escala.

Para comprobar que es posible recoger y utilizar gran cantidad de datos de los vehículos en periodos cortos de tiempo se expondrán diferentes trabajos relacionados con:

- Las capacidades de la nube en cuanto a la comunicación y transmisión de datos.
- Las capacidades de la nube para el tratamiento y aprovechamiento de los datos.
- Arquitecturas en la nube que posibilitan la creación de servicios que utilicen estos datos.
- Arquitecturas centradas en la escalabilidad.
- Arquitecturas en la nube centradas en la privacidad y en la seguridad de los usuarios.
- Arquitecturas que hacen uso de la nube y el big data para proporcionar valor a gran cantidad de datos.

En 2017 el trabajo “*Vehicular network as business model in Big Data, In Intelligent Vehicular Networks and Communications*” [7] de los autores Anand Paul, Naveen Chilamkurti, Alfred Daniel y Seungmin Rho expone un modelo de negocio que contiene una arquitectura de la cuál, se recogen y procesan más de 10TB de datos por hora. Esto muestra la posibilidad de arquitecturas complejas con algoritmos que filtran los datos y los almacenan en bases de datos centralizadas y distribuidas para su computación tanto en tiempo real como en diferido. La aplicación concreta de este documento es de analizar el flujo de vehículos en un determinado lugar. Este trabajo abre la puerta a arquitecturas grandes que posibilitan el tratamiento de muchos vehículos a la vez.

Los investigadores Y. Xu y A. Helal en el trabajo “*Scalable Cloud-Sensor Architecture for the Internet of Things*” [8] en 2016 introducen un sistema en la nube escalable basado en información recopilada por sensores. En este trabajo la nube realiza parte del procesamiento, el que está relacionado con servicios (capa que proporciona la abstracción necesaria sobre capas inferiores para que pueda ser usado por desarrolladores para crear servicios), ejecución (donde se ejecutan los servicios creados), disponibilidad (se crean instancias para la comunicación con las capas inferiores). En esta aplicación se nos muestra la necesidad de realizar una arquitectura escalable y energéticamente eficiente que pueda funcionar en un mundo donde el número de sensores aumenta muy rápido.

En 2015 los investigadores T. Häberle, L. Charissis, C. Fehling, J. Nahm y F. Leymann en su trabajo *“The Connected Car in the Cloud: A Platform for Prototyping Telematics Services”* [9] hablan sobre cómo se puede realizar una arquitectura en la nube basada exclusivamente en abstraer la complejidad de las comunicaciones y del procesamiento y simplificar así el desarrollo de nuevas aplicaciones por parte de terceros sobre esta arquitectura.

Los investigadores JF Colom, H Mora, D Gil y MT Signes-Pont en el trabajo *“Collaborative building of behavioural models based on internet of things”* [10], en 2017 proponen un framework orientado a IoT que permite añadir recursos sin alterar los dispositivos que ya se comunican entre si. Se hace especial mención a permitir que los dispositivos puedan cambiar a trabajar en local o en remoto.

En el trabajo *“Vehicle control system coordinated between cloud and mobile edge computing”* [11] en 2016 por los investigadores K. Sasaki, N. Suzuki, S. Makido y A. Nakao hablan sobre un problema en el que se está actualmente trabajando y se busca una solución, el problema de la latencia. De este artículo podemos extraer que la latencia en la comunicación es clave para ofrecer mejores servicios. En este trabajo podemos ver una solución híbrida en la que la nube forma una parte en el procesamiento de datos pesados.

En 2015 mediante el trabajo *“A computational method based on radio frequency technologies for the analysis of accessibility of disabled people in sustainable cities”* [12], los investigadores V Gilart-Iglesias, H Mora, R Pérez-delHoyo y C García-Mayor proponen un método computacional para identificar problemas de accesibilidad en una ciudad mediante tecnologías basadas en radio frecuencia. Se mezcla el uso de la nube, sensores inteligentes y tecnologías RFID para lograrlo.

Los investigadores H Mora, V Gilart-Iglesias, R Pérez-del Hoyo y MD Andújar-Montoya en 2017 con el trabajo *“A Comprehensive System for Monitoring Urban Accessibility in Smart Cities”* [13] continúan la línea de investigación de [12]. De forma que se busca obtener conocimiento de forma dinámica de problemas de accesibilidad en entornos urbanos. Se busca obtener la información y ponerla en manos de servicios que permitan la toma de decisiones. Se consigue implementar un sistema centrado en los usuarios.

Los investigadores M. Gerla, E. K. Lee, G. Pau y U. Lee en 2014 con el trabajo *“Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds”* [14] exponen una solución basada en la nube y redes de vehículos que se comunican entre si desarrollado sobre el cálculo de rutas en servicios de mensajería y paquetería.

El autor G. S. Tewolde en el trabajo *“Sensor and network technology for intelligent transportation systems”* [15] en 2012 resuelve el problema de forma similar a lo visto en la referencia [14] en este caso más centrado en generar una red que permita la comunicación y el intercambio de datos entre sensores y vehículos.

En 2017 los autores H Mora, JF Colom, D Gil y A Jimeno-Morenilla en *“Distributed computational model for shared processing on Cyber-Physical System environments”* [16] desarrollan un modelo computacional distribuido para compartir carga dinámicamente en diferentes nodos en Cyber-Physical Systems. Abarca el problema del tratamiento de datos distribuyendo la carga entre numerosos dispositivos y usando la nube solo cuando es necesario.

En el 2012 el autor M. Gerla en el trabajo “*Vehicular Cloud Computing*” [17] abarca el problema de la seguridad y la privacidad al usar la nube como forma de procesamiento de los datos recogidos en el vehículo.

En 2012 los investigadores Y. T. Yu, T. Punihaole, M. Gerla y M. Y. Sanadidi en el artículo “*Content routing in the Vehicle Cloud*” [18] se investiga sobre como la nube puede dar soporte a otras tecnologías como V2I (Vehicle-to-Infraestructure) y V2V (Vehicle-to-Vehicle). Trata además el tema de la latencia utilizando una jerarquía que divide los datos según su popularidad para reducir el coste de descubrimiento de contenido entre clusters. Otro artículo que concluye que un enfoque híbrido mejora tiempos de latencia, pero se desconoce la escalabilidad del sistema.

Y. Simmhan en 2013 en su trabajo “*Cloud-Based Software Platform for Big Data Analytics in Smart Grids*” [19] muestran una plataforma en la nube que permite aplicar análisis big data y que abarca una amplia variedad de servicios y de sistemas de los cuales recoger los datos, entre los que están los vehículos. La arquitectura del sistema se basa en recoger una gran cantidad de datos provenientes de sensores, datos del entorno y datos proporcionados por empresas o personas, almacenar y compartir los datos y realizar desarrollos y pronósticos que saquen provecho de los datos.

En 2016 los investigadores Z. Jiao, H. Ding, M. Dang, R. Tian y B. Zhang en el trabajo “*Predictive Big Data Collection in Vehicular Networks: A Software Defined Networking Based Approach*” [20] muestran un Sistema de recolección de datos en redes de vehículos que se centra en la recogida y entrega de los datos cuando estos son un gran número. Se basa en una mezcla de cooperación entre redes móviles y redes ad hoc.

En 2006 con el trabajo “*Real-time arithmetic unit*” [21], los investigadores H Mora-Mora, J Mora-Pascual, JM García-Chamizo y A Jimeno-Morenilla tratan el tema del procesamiento en tiempo real en sistemas de bajo nivel. Se diseña una función aritmética mejorando cálculos predictivos.

El trabajo “*Cloud-Based Vehicular data Analytics Platform*” [22] en 2016 de los investigadores Shashika Ranga Muramudalige y H.M.N. Dilum Bandara muestran un modelo en la nube para el análisis de datos OBD-II. La arquitectura se divide en una capa agente donde se recogen los datos (OBD-II, gps y otros sensores), capa de comportamiento (monitorización del vehículo), capa de análisis (aplicar técnicas de negocio y machine learning) y finalmente un portal web o móvil donde poder visualizar todos los datos.

Mediante estos trabajos relacionados con el uso de la nube y la cantidad de datos que generan los vehículos podemos extraer numerosas conclusiones:

- La latencia es un problema todavía sin resolver. Existen diferentes alternativas de procesamiento de datos que implican la nube y “*Edge computing*” que mejoran los tiempos de respuesta, pero no está del todo claro la escalabilidad de estos sistemas.
- Cuando el volumen de datos aumenta el uso de las capacidades de arquitecturas en la nube es necesario para poder abarcar y procesar esa cantidad de datos.

- Es posible e interesante crear plataformas en la nube que permitan crear servicios a terceros a través de los datos recogidos ya sean de vehículos o de diversos sensores.
- La privacidad y la seguridad son problemas tan importantes como reales que el uso de la nube y la gran cantidad de datos que se obtienen de los vehículos que hay que tener en cuenta y desarrollar arquitecturas que tengan un foco en ellos.

Por lo tanto, para finalizar, indicar que es posible tratar grandes cantidades de datos recogidos de vehículos a través de OBD-II lo que proporciona la posibilidad de que los datos recogidos y el tratamiento de los mismos en este trabajo se puedan realizar a mayor escala desarrollando arquitecturas específicas para estos fines y utilizando la capacidad de computación de la nube.

# Definición redes neuronales artificiales, mapa autoorganizado (SOM)

Un mapa autoorganizado [23] o SOM es un tipo de red neuronal artificial y difiere de otras redes neuronales en cuanto a que este tipo usa una función de vecindad, que permite mantener las propiedades topológicas, del espacio de entrada. Este tipo de red se entrena mediante aprendizaje no supervisado y produce un mapa.

Este tipo de red se compone de neuronas o nodos. A cada uno de estos nodos viene asociado un vector de pesos que tiene la misma dimensión de los vectores de entrada y una posición en el mapa.

La forma de aprendizaje de un mapa autoorganizado es que ciertos patrones de entrada provoquen que lugares de la red respondan de forma similar.

Esta red utiliza aprendizaje competitivo que es una forma de aprendizaje usada en aprendizaje no supervisado. En este tipo de aprendizaje los nodos compiten por tener el derecho para responder a un conjunto de datos de entrada. Este tipo de aprendizaje es muy adecuado para encontrar clusters dentro de datos.

El aprendizaje competitivo se compone de tres elementos básicos:

- Conjunto de neuronas en los que son iguales todas, excepto algunos pesos que se distribuyen aleatoriamente. Esto hará que estas respondan de manera diferente a los mismos conjuntos de patrones de entrada.
- Un límite impuesto a cada neurona.
- Un mecanismo que permite que las neuronas compitan por responder a los datos de entrada, de modo que únicamente una neurona de salida por grupo esté activada a la vez.

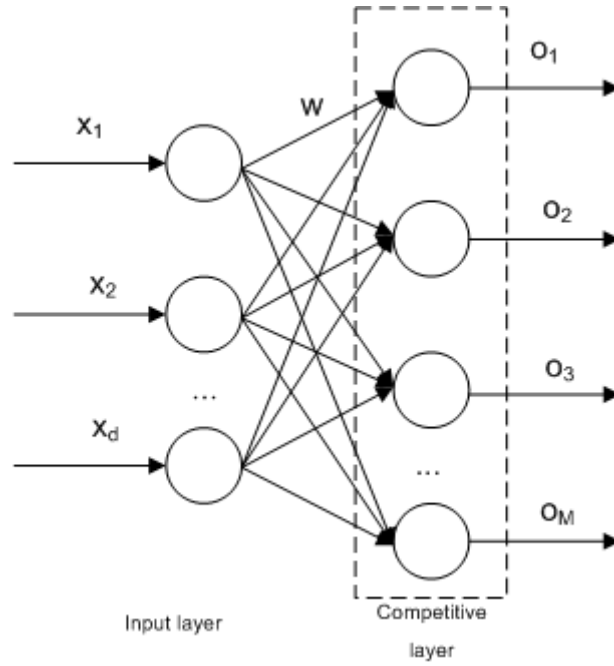


Imagen 2: Arquitectura red aprendizaje competitivo [24]

Donde cada neurona competitiva tiene un vector de pesos tal que:

$$w_i = (w_{i1}, \dots, w_{id})^T, i = 1, \dots, M$$

Los datos de entrada son:

$$x^n = (x_{n1}, \dots, x_{nd})^T \in \mathbb{R}^d$$

Y la salida:

$$o_i = 0, i = 1, \dots, M, i \neq m$$

Concluyendo con la explicación del mapa autoorganizado, la neurona que tiene el vector de pesos más parecido a la entrada es denominada BMU o unidad de mejor correspondencia. Finalmente, la fórmula utilizada para actualizar el vector de peso de una neurona es la siguiente:

$$W_v(s+1) = W_v(s) + \theta(u, v, s) \alpha(s) (D(t) - W_v)$$



Dentro de esta fórmula podemos encontrar:

- $s$ , índice del paso.
- $t$ , índice dentro del conjunto entrenante.
- $u$ , índice de la unidad de mejor correspondencia
- $D(t)$ , vector de entrada.
- $\alpha(s)$ , coeficiente monótonamente (función monótona, función entre conjuntos ordenados que preserva o invierte el orden dado) decreciente de aprendizaje.
- $\theta(u,v,s)$ , función de vecindad, la cual depende de la distancia entre la unidad de mejor correspondencia y la neurona  $v$ . Esta puede obtener un valor de 1 o 0 según la cercanía con la BMU o elegir una función gaussiana.

Este proceso se repite para cada vector de entrada un número  $\lambda$  (ciclos), generalmente grande.

Por último, calculando la distancia euclidiana entre el vector de entrada y los de pesos se descubrirá la única neurona ganadora, que es aquella que se encuentra más cerca del vector de entrada.

## Algoritmo SOM

En este apartado se expondrá el algoritmo habitual que describe el funcionamiento de un mapa autoorganizado:

1. Realizar un mapa de neuronas en el que sus pesos sean aleatorios.
2. Coger un vector de entrada igual a  $D(t)$ .
3. Iterar por cada neurona en el mapa.
  - a. Usar la fórmula de la distancia euclídea para encontrar similitudes entre el vector de entrada y los vectores de pesos de las neuronas del mapa autoorganizado.
  - b. Seguir los nodos que producen las distancias más pequeñas, es decir la unidad de mejor correspondencia BMU.
4. Actualizar los nodos vecinos de la unidad de mejor correspondencia, incluyendo a esa misma unidad.
  - a.  $W_v(s+1) = w_v(s) + \theta(u,v,s) \times \alpha(s)(D(t) - W_v)$
5. Incrementar  $s$  y volver al paso 2, mientras  $s < \lambda$

*Imagen 3: Algoritmo mapa autoorganizado[23]*

## Interpretación SOM

En este apartado se busca responder a como se interpreta los resultados obtenidos a partir de un mapa SOM.

Un mapa autoorganizado se puede interpretar de dos formas diferentes:

- De forma que un mapa SOM forme un mapa semántico donde las muestras que tengan similitudes sean mapeadas juntas, y las que sean diferentes entre si sean mapeadas aparte. Esto se debe a cuando en la fase de entrenamiento los pesos de las unidades vecinas son movidos en la misma dirección. Esto provoca que los elementos que son similares exciten (acerquen) a neuronas adyacentes.
- De forma que un mapa SOM se interprete como un mapa de regiones donde los pesos de las neuronas actúan como punteros al espacio de entrada. Con esta aproximación existirán más punteros (pesos de neuronas) en regiones donde existen más muestras de entrenamiento y menos donde existen menos.

## Trabajos previos relacionados sobre mapas autoorganizados

Para trabajar con los datos obtenidos de los vehículos y clasificarlos correctamente vamos a utilizar los mapas autoorganizados. En este apartado se va a mostrar como otros trabajos han realizado pruebas sobre este tipo de red neuronal artificial y han mostrado numerosas aplicaciones y utilidades.

En 2017 los investigadores G. G. Colombini, I. B. M. de Abreu y R. Cerri en el trabajo “*A self-organizing map-based method for multi-label classification*” [25] utilizan el mapa autoorganizado para resolver un problema de “*multi-label classification*”. Un sistema “*multi-label*” es aquel en el que las instancias pueden ser clasificadas dentro de muchas clases simultáneamente. Este problema suele estar orientado a métodos de aprendizaje supervisado. El trabajo consiste en entrenar instancias que son mapeadas a las neuronas de una red que se organiza así misma realizando el mismo proceso. Los experimentos realizados en este trabajo muestran resultados prometedores comparados con técnicas habituales como el aprendizaje supervisado. Este trabajo nos muestra la capacidad de este tipo de red.

En el trabajo “*A connectionist computational method for face recognition*” [26] 2016 los investigadores FA Pujol, H Mora y JA Girona-Selva desarrollan un método para el reconocimiento facial mezclando una modificación del algoritmo EBGM y redes SOM. En este caso podemos ver como las redes SOM son muy flexibles y una buena opción para clasificar.

Los investigadores J. Faigl y G. A. Hollinge en el trabajo “*Autonomous Data Collection Using a Self-Organizing Map*” [27] resuelven el problema del viajero (TSP) para desarrollar una solución a la recolección de datos autónoma. En este trabajo se puede ver la capacidad de adaptación de este tipo de red. Los investigadores varían el mapa autoorganizado de forma que se pueda extender los casos de uso de esta red cuando las distintas fuentes de datos no son fiables al cien por cien.

Los investigadores K. Okada, K. Matsui, J. Haase y H. Nishi en 2015 con el trabajo “*Privacy-preserving data collection for demand response using self-organizing map*” [28] se propone un método para dotar de privacidad a datos mediante el uso de una red SOM. El problema tratado es publicar datos de consumo de energía sin comprometer la privacidad de quienes consumen esa energía (se habla de picos de consumo y como estos pueden delatar cuando el usuario llega a casa).

En el trabajo “*An optimized cloud based big data processing mechanism using Self-Organizing Map in Hadoop environments*” [29] de 2015 los investigadores G. N. Iyer, S. Silas y G. Iyer se trata el problema de algoritmos de optimización de búsqueda bajo entornos de gran amplitud. Utilizando SOM se consigue una mejor velocidad que con otras metodologías. Los problemas que podrían ser tratados son de alto coste computacional como problemas de procesamiento de imagen o de física. Este trabajo nos muestra como mapas autoorganizativos pueden ser usados para procesar o para optimizar flujos de trabajo en entornos complejos.

En 2016 el investigador R. Lasri en su trabajo “*Clustering and classification using a self-organizing MAP: The main flaw and the improvement perspectives*” [30] se centra en el SOM como herramienta para clasificar y agrupar datos. En este trabajo se centra en medir el rendimiento del SOM para estas tareas y proponer mejoras. Este trabajo concluye indicando que el mapa autoorganizativo tiene muchas ventajas y presenta un buen rendimiento frente a otros tipos de procesos de clasificación. Por otro lado, sus puntos débiles residen en la normalización del espacio de entradas, donde este pierde algo de precisión y las neuronas no pueden diferenciar las entradas originales. Sin embargo, no es un problema que no se pueda resolver, en estos casos hay que utilizar herramientas que manejen el espacio de entrada son afectar a la habilidad de clasificación.

En 2016 J Azorin-Lopez et al, en el trabajo “*Constrained self-organizing feature map to preserve feature extraction topology*” [31] expone una red SOM que permite preservar la topología de localizaciones cuando un vector es extraído. De forma que, se permite mejorar la información obtenida de los mapas SOM.

En 2016 los investigadores D. I. Kumar y M. R. Kounte en el trabajo “*Comparative study of self-organizing map and Deep self-organizing map using MATLAB*” [32] se realiza un estudio para testear la eficiencia de los mapas autoorganizados y una variante de los mismos, denominados DSOM o mapas autoorganizados profundos, para catalogar imágenes.

Con estos trabajos estudiados del uso de redes SOM podemos sacar las siguientes conclusiones:

- Las redes SOM son capaces de resolver problemas de clasificación.
- Las redes SOM permiten utilizar aprendizaje no supervisado en problemas de alto coste computacional como el tratamiento de imágenes.
- Las redes SOM con algunas variaciones también pueden ser útiles cuando no se dispone de flujo constante de información.
- Otras variaciones permiten usar estas redes para dotar de privacidad a conjuntos de datos.

Para concluir, indicar que este tipo de red tiene ciertas características, como el aprendizaje no supervisado, que son interesantes para la aplicación de este trabajo. Este tipo de aprendizaje se utiliza cuando no hay un conocimiento a priori sobre los datos a tratar. Por otro lado, se ha podido ver como este tipo de redes no tiene limitaciones claras y se puede usar en un gran número de casos.

# Objetivos

Los objetivos de este trabajo son los siguientes:

## OBJETIVO 1:

Realizar una investigación del estado del arte sobre los diferentes diseños, arquitecturas y evolución sobre los sistemas de conducción inteligente.

## OBJETIVO 2:

Identificar las partes de los sistemas de conducción inteligente tales como, recogida de datos OBD-II, tratamiento de datos y posibilidades y estados de la conducción.

## OBJETIVO 3:

Propuesta de un diseño que permita una vez obtenidos datos relacionados con la conducción, procesar los mismos y determinar en cierto grado la calidad de la conducción.

## OBJETIVO 4:

Implementar un prototipo de ese diseño que ponga en práctica los conocimientos adquiridos del estado del arte y establezca las bases para el desarrollo precompetitivo.

# Cuerpo del trabajo

## Diseño de la arquitectura

### Parte 1. Vector de datos

#### Objetivos y método

En esta primera parte se pretende adaptar un vector de datos recogido mediante el puerto OBD-II de un vehículo. El puerto OBD-II es un sistema que nos permite recoger información del vehículo en tiempo real. Los datos proporcionados sirven en primer lugar como monitorización y control del vehículo.

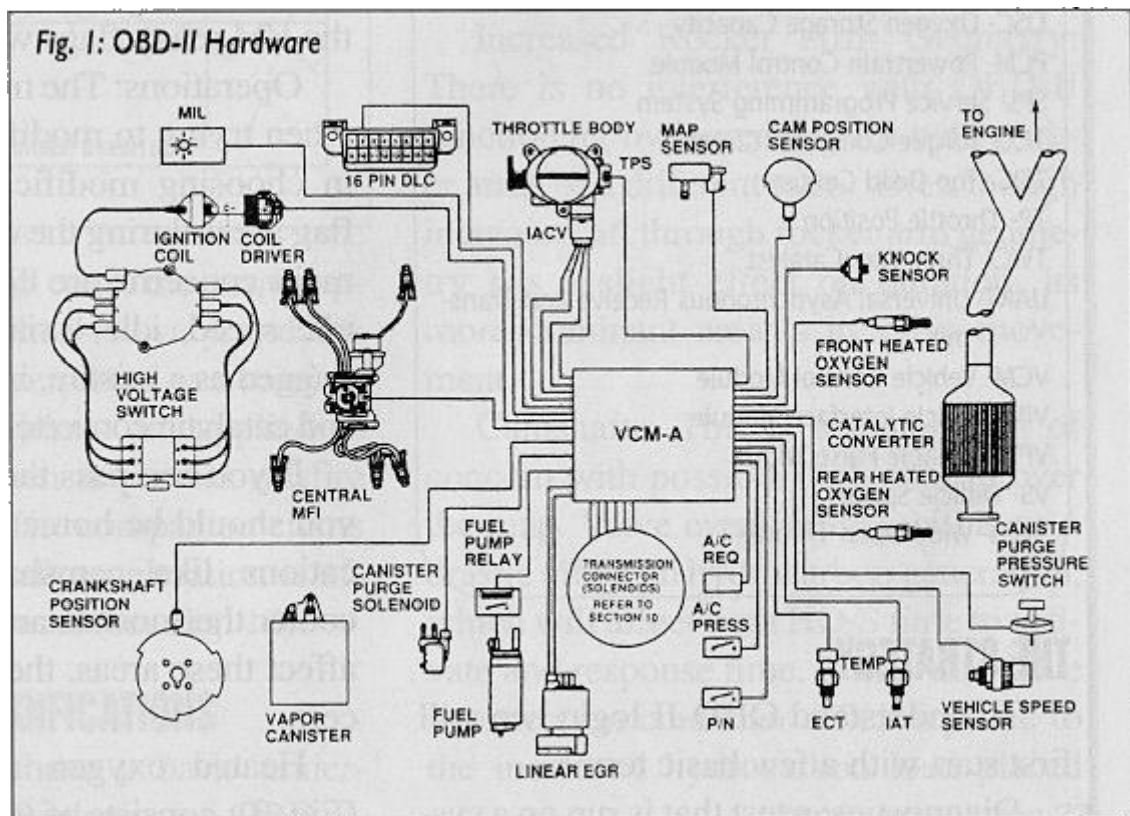


Imagen 4: Ejemplo de sensores y OBD-II [33]

Una vez se han obtenido estos datos se deberá elegir cuales de estos van a servir para la aplicación que corresponde a este trabajo. Por otro lado, se tendrá que adaptar estos datos para su correcta utilización en la red neuronal.

Finalmente, habrá que organizar dichos datos y dotarlos, tanto de una línea temporal como de un único usuario. Este proceso será de gran importancia puesto que para conocer el comportamiento a lo largo de una conducción esta deberá estar organizada en el tiempo y no podrá ser de conductores diferentes.

## **Obtención de datos**

La obtención de datos se realizaría a través del sistema OBD-II de diagnóstico de un vehículo compatible. Se realizarían numerosas pruebas que permitan discernir propiedades como el estado del conductor, la calidad de la conducción (suave, medio, agresivo), el lugar donde se realiza la conducción (tipo de vía o calidad de la misma) y el tiempo inicial y final de la conducción.

En este diseño se presupondrá la recogida de datos y se dispondrán de datos ya recogidos correctamente, dado que la problemática que supone recoger esos datos y realizarlo con un número grande de muestras escapa de este trabajo.

## **Datos utilizables**

El sistema de diagnóstico OBD-II es capaz de recoger una gran cantidad de datos. Estos datos se recogen a través de “*Parameter IDs*” y viene determinado por el modo de operación en el que se configure.

Modo (hex)	Descripción
01	Mostrar datos actuales
02	Mostrar datos de un marco congelado
03	Mostrar códigos de problemas de diagnóstico almacenados
04	Borrar códigos de diagnóstico y valores almacenados
05	Resultados del test del sensor de oxígeno
06	Resultados del test de otros componentes
07	Mostrar códigos de problemas de diagnóstico pendientes
08	Funcionamiento del sistema abordo
09	Solicitar información del vehículo
0A	Códigos de problemas de diagnóstico permanentes

*Imagen 5: Descripción de modos disponibles OBD-II [34]*

Vistos estos modos de funcionamiento, el más indicado para esta aplicación es el primero de ellos, el que corresponde al hexadecimal 01. Este modo nos permite acceder a una gran cantidad de datos de los cuales se seleccionarán los más indicados para la aplicación. La tabla descriptiva con los datos que se utilizarán se muestra a continuación:



PID (hex)	Bytes devueltos	Descripción	Val. Mín	Val. Max	Unidades	Fórmula
04	1	Carga del motor	0	100	%	$\frac{100}{255}A$
05	1	Temperatura del refrigerante del motor	-40	215	°C	$A - 40$
0C	2	RPM del motor	0	16383,75	Rpm	$\frac{256A + B}{4}$
0D	1	Velocidad del vehículo	0	255	Km/h	$A$
0F	1	Temperatura en la toma de aire	-40	215	°C	$A - 40$
49	1	Posición pedal del acelerador D	0	100	%	$\frac{100}{255}A$

Imagen 6: Descripción de valores útiles [34]

## Requisitos funcionales

Los requisitos funcionales que debe tener la recolección y la selección de datos es el siguiente:

### REQUISITO FUNCIONAL 1:

Los datos deben ser recogidos mediante el sistema OBD-II en el modo operacional 01, dentro de este modo los datos a recoger serán los correspondientes al PID 04, 05, 0C, 0D, 0F, 49.

### REQUISITO FUNCIONAL 2:

El vector de datos debe contener una marca temporal junto a un identificador único.

### REQUISITO FUNCIONAL 3:

El identificador corresponderá a un único trayecto de conducción realizado por un conductor.

## Parte 2. Almacenaje de datos

### Objetivos y método

El objetivo de la parte 2 es elaborar un plan una vez disponemos de los vectores de datos. Estos datos tendrán que ser adaptados y almacenados para conseguir un correcto funcionamiento de la red neuronal artificial y una posterior clasificación sin problemas.

En primer lugar, los datos deberán ser adaptados para su posterior utilización en la red neuronal. El método utilizado para esto corresponde a la normalización estadística. Este proceso se realiza al disponer de datos que comparten escalas diferentes.

Sin embargo, hay que tomar ciertas precauciones a la hora de normalizar datos si estos van a ser tratados por la red neuronal. Este proceso de normalización podría arruinar el posterior análisis de los mismos y su resultado. Por lo que se debe estar seguro de la normalización realizada [35].

Una vez los datos están normalizados, estos deben ser organizados en un formato que permita su posterior extracción. El vector debe quedar organizado de forma que se identifique por el identificador único de forma que los demás datos puedan ser empleados para el análisis. El dato de tiempo estará junto a los datos correspondientes al obtenido en el OBD-II de forma que se pueda diferenciar la misma conducción en diferentes estados de tiempo.

ID	Tiempo	Carga del motor	RPM	Velocidad	Acelerador	Temp. Toma Aire
1	02/02/2017/10:00:00	20	10000	58	30	80

*Imagen 7: Ejemplo de organización de los datos en la BD*

## **Características del almacenaje de datos**

Para almacenar estos datos utilizaremos archivos del tipo “csv” o valores separados por comas. Este tipo de documento tiene un formato sencillo que permite representar los datos en formato de tabla. El funcionamiento base consiste en indicar las columnas de la tabla a través de comas en una misma línea y las distintas filas de la tabla representarlas como saltos de línea en el archivo. De esta forma tenemos la información almacenada como si de una base de datos se tratará y además estará lista para ser procesada.

No será necesario implementar ninguna tecnología de base de datos puesto que para tratar datos en formato “csv” existen numerosas librerías para lenguajes de programación o programas como “Matlab” que pueden trabajar directamente con dicho formato.

## **Requisitos funcionales**

### **REQUISITO FUNCIONAL 1:**

El sistema debe ser capaz de recoger datos obtenidos previamente y en formato vector de datos.

### **REQUISITO FUNCIONAL 2:**

La tabla debe contener valores para ID, tiempo, carga del motor, temperatura del refrigerante del motor, RPM, velocidad, temperatura en la toma de aire y posición pedal del acelerador D.

### **REQUISITO FUNCIONAL 3:**

El sistema debe permitir comunicación y enviar datos hacia la red neuronal SOM.

## **Requisitos no funcionales**

### **REQUISITO NO FUNCIONAL 1:**

El sistema debe tener la capacidad de escalar.

### **REQUISITO NO FUNCIONAL 2:**

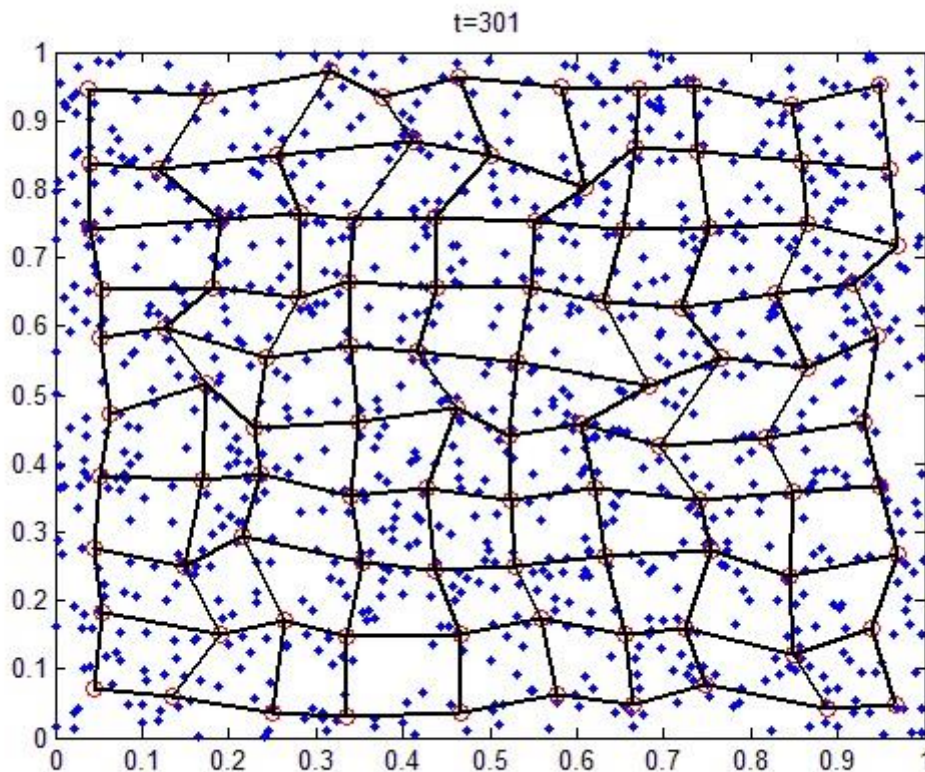
El sistema debe tener la capacidad de funcionar en la nube bajo entornos de alta carga.

## Parte 3. Tratamiento de datos

### Objetivos y método

Una vez tenemos todos los datos organizados y ordenados según los criterios establecidos, se tendrá que realizar su tratamiento. Para poder dotar de utilidad y otorgar una clasificación se aplicará una red neuronal artificial.

Esta red neuronal será la encargada de proporcionarnos visualización a nuestros datos. La función en definitiva consistirá en mapear datos multidimensionales en subespacios de menor dimensión donde las relaciones geométricas entre puntos indican similitud [29, 30, 31]. En la siguiente imagen se puede ver un ejemplo de cómo diferentes datos forman relaciones geométricas:



*Imagen 8: Imagen ejemplo de mapa autoorganizado [39]*

## Red neuronal utilizada

Como se ha ido avanzando a lo largo del trabajo la red neuronal artificial empleada en este diseño será una red neuronal autoorganizada.

En primer lugar, una vez estudiado el problema, se encuentra que estamos ante un problema de clasificación de datos. Dispondremos de un conjunto de datos extraídos de vehículos, realizando trayectos, y, como se ha indicado en el objetivo tres del trabajo, queremos clasificarlos para obtener información que no podemos ver a partir de los datos en bruto. Debido al desconocimiento de cómo tratar los datos correctamente y de no saber cómo estos datos se relacionan entre sí, la solución propuesta para tratar este problema es utilizar una red neuronal, que no es más que un modelo computacional con un comportamiento similar al que realizan las neuronas en un cerebro biológico. Las redes neuronales, debido a sus características y a sus similitudes con el cerebro son muy buenas para resolver problemas diversos como: aproximación de funciones, procesamiento de datos, reconocimiento de patrones y clasificación.

Características y propiedades como neuronas, conexiones entre neuronas y pesos de neuronas permiten a la red establecer patrones de la información dada.

Dentro de las redes neuronales, se ha elegido un mapa autoorganizado. Este tipo de mapas mediante su funcionamiento (las neuronas que procesan los datos tienen asociado vectores de pesos de la misma dimensión que los vectores de entrada), se hacen idóneos para bajar la dimensionalidad (representar en un mapa que a simple vista nos proporcione información que no podíamos ver anteriormente) a datos de altas dimensiones (datos recogidos por los vehículos).

Gracias a sus características como el aprendizaje no supervisado y la reducción de dimensionalidad permite interpretar datos que de otra forma no parecerían interpretables.

La forma de organización de esta red es sencilla, radica básicamente en ajustar los datos entrantes mediante una función de proximidad hacia las neuronas ganadoras. Es decir, según los datos nuevos se parezcan a la entrada de una muestra (aprendizaje competitivo).

Por lo que se ha podido comprobar en el estado del arte previo, esta red neuronal artificial es útil cuando tenemos un problema de “*clustering*” o clasificación de datos. Por lo tanto, es muy válido para esta aplicación.

## Problemas mapa autoorganizado

El principal inconveniente de un mapa autoorganizado es que requiere que los pesos de las neuronas sean necesarios y suficientes para realizar la agrupación. Si se proporciona un mapa autoorganizado con muy poca información o demasiada información diferente en los pesos, las agrupaciones que se dibujan en el mapa pueden no ser lo suficientemente exactas como para ser válidas. Esto quiere decir, que los vectores de peso deben basarse en datos que puedan agrupar y distinguir con éxito las entradas.

Un segundo problema con los SOM es que a veces es difícil obtener un mapeado perfecto donde las agrupaciones son únicas dentro del mapa. Por ejemplo, las anomalías aparecen en el mapa como dos grupos similares en diferentes áreas. Los “*clusters*” se dividen en “*clusters*” más pequeños creando áreas de neuronas similares. Para resolver este problema hay que prestar atención a la inicialización del mapa.

Como último problema, los mapas autoorganizados requieren que los puntos cercanos se comporten de manera similar. Problemas como la puerta XOR no tienen esta propiedad y no pueden tener una asignación estable en un mapa autoorganizado.

## **Ventajas mapa autoorganizado**

La principal ventaja que tenemos al usar un mapa autoorganizado es que los datos serán fáciles de interpretar y de entender. La reducción de la dimensionalidad y las características de “*clustering*” hace que sea fácil observar similitudes en los datos. Se hace posible alterar parcialmente piezas de datos para tener un efecto en la entrada del sistema.

Los mapas autoorganizados son capaces de manejar numerosos tipos de problemas de clasificación, al mismo tiempo que proporcionan un resumen inteligible de los datos proporcionados. Por ejemplo, existen SOM que son utilizados como parte de herramientas de búsqueda o filtros spam. Utilizando vectores de pesos adecuados los mapas autoorganizados son capaces de adaptarse.

Finalmente, los SOM son capaces de agrupar conjuntos de datos grandes y complejos, de forma que, con unas pocas técnicas de optimización, este se puede entrenar en un periodo corto de tiempo. Además de ser un algoritmo fácil de entender como para que el proceso se pueda alterar según sea necesario.

## **Requisitos funcionales**

### **REQUISITO FUNCIONAL 1:**

La red neuronal artificial que se ha de utilizar es un mapa autoorganizado.

### **REQUISITO FUNCIONAL 2:**

La red neuronal debe ser capaz de realizar un “*clustering*” de los datos recogidos por OBD-II

### REQUISITO FUNCIONAL 3:

El resultado final del mapa autoorganizado nos debe proporcionar visibilidad sobre los datos que hemos introducido.

## Requisitos no funcionales

### REQUISITO NO FUNCIONAL 1:

Se deben configurar las diferentes entradas de forma que los vectores de peso deben basarse en datos que puedan agrupar.

## Parte 4. Clasificación de datos

### Objetivos y método

Después de que la red neuronal actúe, dispondremos de un “mapa” en el que los datos serán representados, formando diferentes conjuntos comunes. Este mapa nos dejará ver diferentes patrones entre los datos que de otra forma sería imposible y permitirá dotar de aplicaciones a los mismos.

En este diseño buscamos que los datos nos proporcionen características de la conducción que se está produciendo. Es decir, queremos que los datos recogidos al principio del diseño nos sirvan para determinar las características de la conducción realizada.

Por lo tanto, el primer objetivo fundamental de esta parte es determinar cuáles son las características que queremos descubrir y cuáles pueden ser obtenidas a través de la información que extraemos del uso del mapa autoorganizado.

Las características seleccionadas deben poder ser extraídas del mapa y deben contener información útil para la mejora en el proceso de la conducción. Las conclusiones de esta clasificación deben aportar algún valor a la conducción en forma de aumento de la seguridad o eficiencia energética.

Una vez determinamos las características, se elegirá las distintas variaciones entre las que estas pueden oscilar. Es decir, clasificar un determinado parámetro de la conducción dentro de patrones como podrían ser suave o agresivo.

Como último paso, se tendrá clasificar los datos obtenidos en una tabla que tenga la forma indicada, en la que a la izquierda podamos observar los patrones de conducción y encima las diferentes características del mismo.

Finalmente, se podrá identificar que datos y características corresponden a un tipo de conducción u otro y determinar la calidad de conducción para datos nuevos de entrada.

Así se lograría el principio de este trabajo. Determinar que parámetros y en qué medida estos corresponden a una conducción óptima o no.

## **Criterio de clasificación**

Los criterios de clasificación los podemos dividir en dos apartados.

En primer lugar, los que corresponden a las características o parámetros de la conducción:

- **Estilo de conducción:** determinará como se está llevando a cabo la conducción. En este parámetro se determinará la tranquilidad o la agresividad con la que el conductor lleva a cabo la tarea. Entre los valores que pueden ayudar a lograr esta clasificación podemos encontrar las revoluciones por minuto del motor o la posición del pedal del acelerador D. Un ejemplo que podemos encontrar en un estilo agresivo es aquel en el que el conductor presiona el acelerador un gran número de veces y en gran porcentaje (ejerce mucha presión en el pedal). Otro ejemplo en el lado opuesto podría ser como una conducción tranquila se caracteriza por un motor que no sufre grandes variaciones.
- **Combustible:** determinará si se está haciendo un uso eficiente del combustible. El uso del acelerador es un muy buen indicador de la cantidad de combustible y de si este se está utilizando de una manera óptima. A mayor presión ejercida en el acelerador mayor uso de combustible y viceversa. Por otro lado, las variables correspondientes a carga del motor y revoluciones por minuto del motor también son buenos indicadores del uso del combustible. Ambos, funcionan de la misma forma que el acelerador, a mayor porcentaje (en caso de carga del motor) o mayor valor de revoluciones por minuto (en el caso de las RPM del motor) mayor gasto del combustible y por lo general peor uso de combustible se está haciendo del mismo.
- **Ruido o contaminación acústica:** determinará si el ruido que está realizando el vehículo es alto o bajo. Este valor podría venir dado a partir de dos variables. La primera velocidad del vehículo, a mayor velocidad del vehículo mayor ruido se genera. La segunda y más notoria las revoluciones por minuto del motor. El motor es la principal fuente de ruido del vehículo (a excepción del uso del claxon que no se puede medir). Si el motor corre a altas revoluciones el ruido del mismo será mayor.
- **Contaminación generada:** determinará el grado de contaminación que desprende el vehículo. Generalmente deriva de la eficiencia del combustible puesto que los parámetros son los mismos y actúan en el mismo grado. El uso del acelerador, las revoluciones por minuto del motor y la carga del motor determinan si se consume más combustible y por tanto se contamina más.



- Estado del vehículo: determinará si existe alguna posibilidad de que el vehículo no funcione correctamente. Este valor vendrá determinado por la temperatura en la toma de aire y la temperatura del refrigerante del motor. Que, si bien no son los únicos indicadores de avería, pueden ser útiles para conocer si el estado del vehículo es el más óptimo.

Por último, el segundo criterio de clasificación es aquel que indica el grado de los elementos de la anterior clasificación. Los valores son los siguientes:

- Normal: indica que se está realizando la conducción de manera normal.
- Ineficiente: indica conducción ineficiente o agresiva. Para los parámetros anteriores:
  - o Estilo de conducción agresivo.
  - o Uso ineficiente del combustible.
  - o Alto grado de contaminación acústica.
  - o Alto nivel de contaminación generada.
  - o Estado del vehículo peligroso.

Una vez se dispone de estos criterios la tabla de clasificación sería como el siguiente ejemplo:

	Estilo	Combustible	Ruido	Contaminación	Estado del vehículo
Normal	RPM estables	Presión del acelerador baja	RPM bajos	Valores de carga del motor bajos	Temperatura del refrigerante del motor baja
Ineficiente	Mucha presión en el pedal del acelerador	Valores de RPM altos	RPM altos	Valores de carga del motor variables	Temperatura del refrigerante del motor errática

*Imagen 9: Tabla ejemplo Parámetros/Características*

## **Requisitos funcionales**

### **REQUISITO FUNCIONAL 1:**

La clasificación de los datos debe aportar valor a los datos recogidos del mapa autoorganizado.

### **REQUISITO FUNCIONAL 2:**

Los parámetros de la conducción corresponden a: estilo, combustible, ruido, contaminación y estado del vehículo.

### **REQUISITO FUNCIONAL 3:**

La conducción se podrá clasificar en normal e ineficiente.

## **Implementación del prototipo**

### **Parte 1. Recogida y preparación de datos**

#### **Elaboración de circuitos**

Para la realización del prototipo es necesario la obtención de datos de vehículos mediante el puerto OBD-II del mismo y una aplicación que permita recoger y almacenar los datos. Para cumplir con este objetivo se han realizado unos circuitos que ejemplarizan diferentes situaciones de una conducción normal. Los circuitos se dividen en:

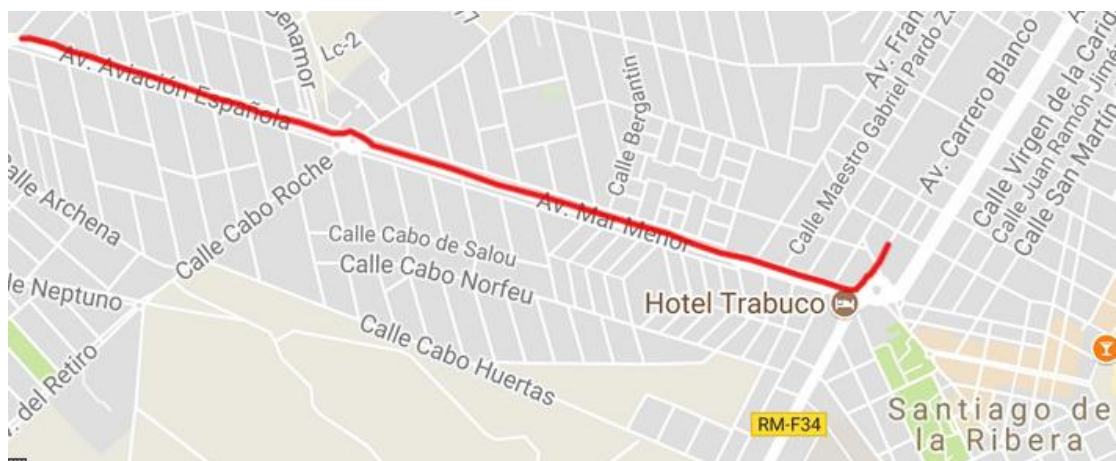
- Conducción rápida: conducción por vías rápidas como puede ser una autovía, autopista o vías rápidas en la ciudad.
- Callejeo: conducción realizada dentro de la ciudad donde aparecen factores que ralentizan la conducción como semáforos, pasos de peatones o tráfico.
- Mixtos: circuitos donde se mezclan situaciones de vía rápida y callejeo por la ciudad.

## Circuitos de conducción rápida

Los circuitos dentro de la conducción rápida tienen las siguientes características:

- Se alcanzan en algún punto los 60 km/h.
- Está compuesto por una vía rápida, a saber, autovía, autopista o avenida.
- No se realizan detenciones durante la conducción.
- El tráfico de la vía tiene que ser fluido.
- Una conducción habitual en este tipo de vías sería suave y la velocidad se mantendría durante parte del trayecto.

Los siguientes dos circuitos han sido seleccionados y probados dado que cumplen las anteriores características.



*Imagen 10: Circuito conducción rápida 1*

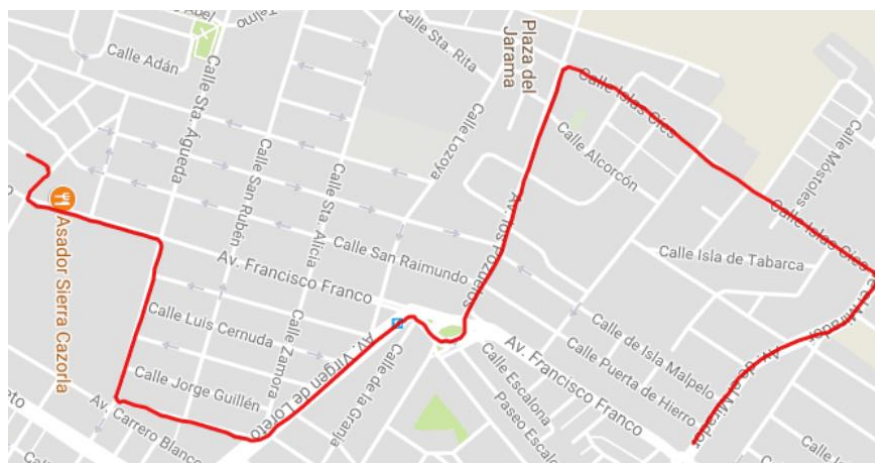


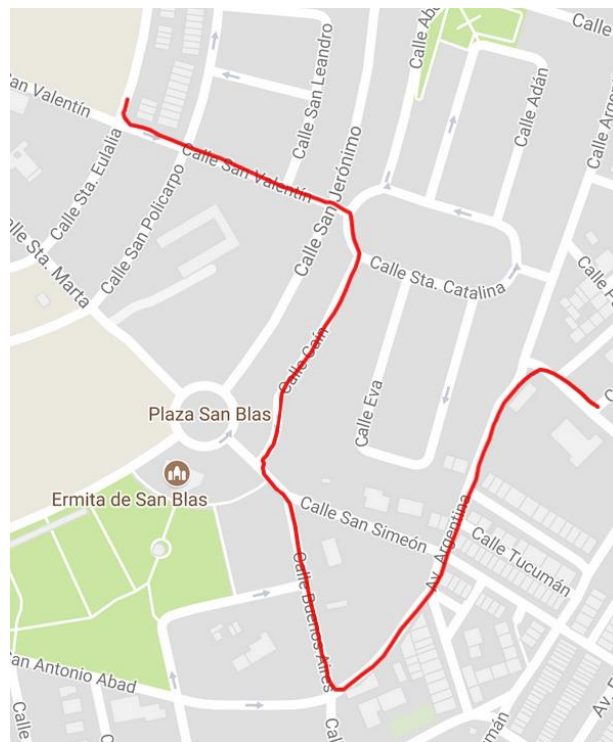
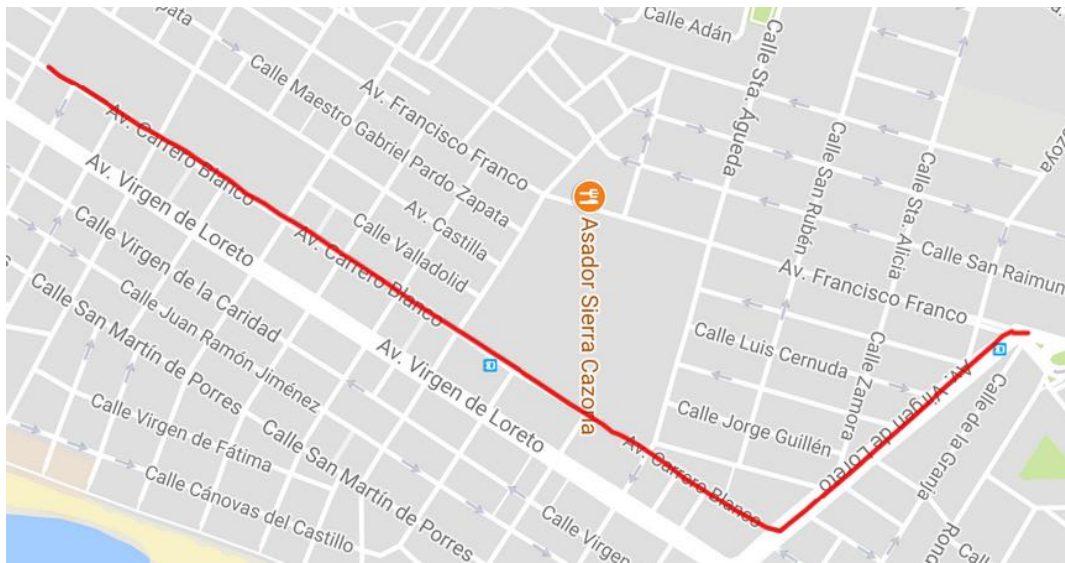
## Circuitos de callejeo

Este tipo de circuitos cuentan con las siguientes características:

- La velocidad máxima que se alcanza es de 50 km/h.
- En los circuitos aparecen semáforos, ceda el paso, pasos de peatones u otros condicionantes que producen que la velocidad durante el trayecto sea variable.
- Solo se producen dentro de la ciudad.

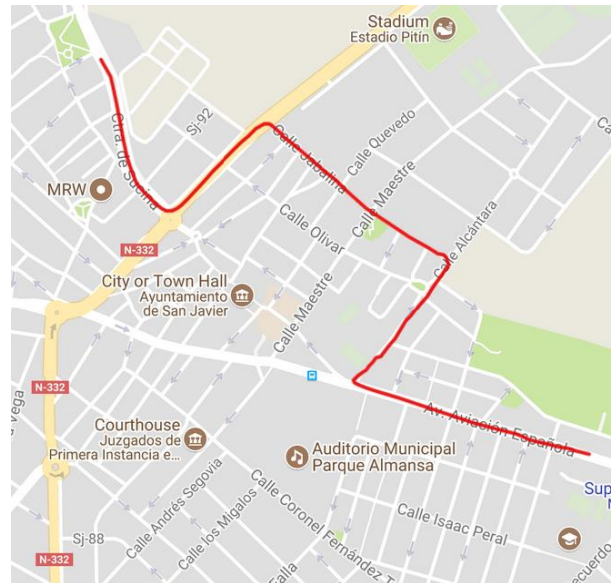
Para completar este apartado, se muestran los tres circuitos seleccionados.



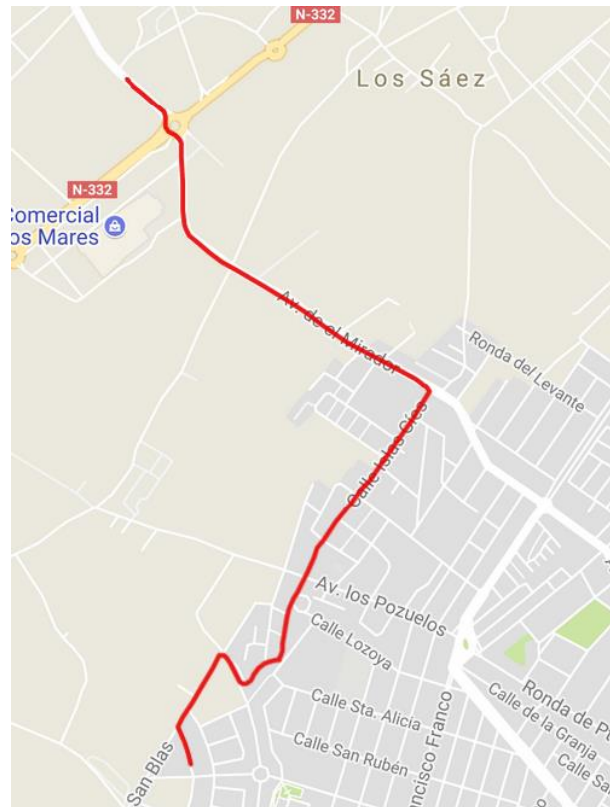


## Circuitos mixtos

Los circuitos mixtos se corresponden a circuitos donde ocurren situaciones de los dos tipos de circuitos anteriores. Es decir, existen tramos del circuito correspondientes a un circuito rápido con sus correspondientes características y tramos con características del circuito del callejeo. Estos circuitos, en definitiva, buscan simular una situación cotidiana de cualquier conductor.



*Imagen 15: Circuito mixto 1*



*Imagen 16: Circuito mixto 2*

## **Vehículo de pruebas**

El vehículo utilizado durante las pruebas es un Citroen C4 Cactus (2014) y cuenta con las siguientes características:

- Peso 1095 kg.
- Tracción delantera.
- Combustible gasolina.
- Potencia 82 cv.
- Par motor 2750 rpm.
- Cambio manual.

## **Parte 2. Red neuronal, mapa autoorganizativo**

### **Análisis de datos recogidos**

En primer lugar, y antes de utilizar el mapa autoorganizativo, se va a analizar los diferentes datos obtenidos. Se va a comparar los diferentes circuitos según los valores de los datos recogidos.

El objetivo de este apartado es comparar los valores de carga, temperatura del refrigerante, RPM, velocidad, temperatura en la toma de aire y posición pedal del acelerador D entre los distintos tipos de circuitos. De esta forma, se podrá observar cuales son los patrones de conducta según el tipo de vía (circuito) que se recorra.

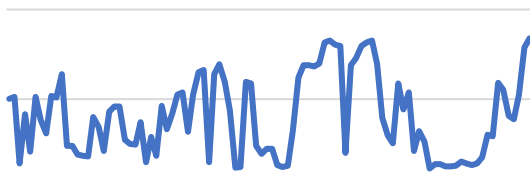
### **Carga del motor**

En cuanto a la carga del motor, en los gráficos mostrados a continuación, se puede observar ligeras diferencias entre los circuitos de conducción rápida y callejeo. La carga del motor muestra valores generalmente más bajos y con mayores variaciones en los circuitos correspondientes a callejear. Los valores bajos pueden indicar como el motor tiene menor carga de trabajo cuando las revoluciones o la velocidad del mismo es baja. En cuanto a los valores variables más frecuentes puede deberse a los sucesivos acelerones y parones que producen el tráfico y los semáforos en los circuitos de calle.

Por otro lado, en la conducción mixta se aprecian variaciones muy similares a las del circuito del callejeo, por lo que en este no se pueden sacar unas conclusiones claras.



### Circuito conducción rápida



### Circuito callejear

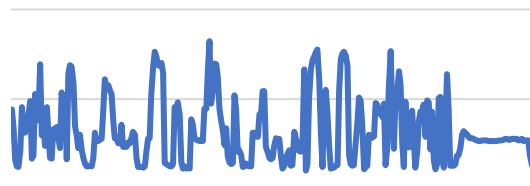


Imagen 17: Tabla comparativa carga del motor conducción rápida vs callejear

### Circuito mixto

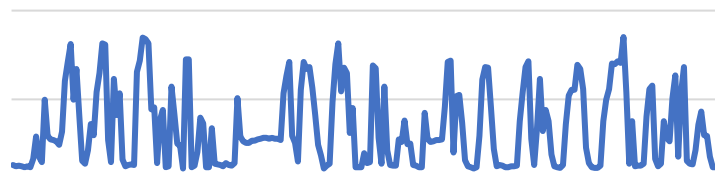


Imagen 18: Tabla comparativa carga del motor conducción mixta

# Temperatura del refrigerante del motor

La temperatura del refrigerante del motor parece no tener variaciones claras en cuanto a los distintos tipos de circuitos. Generalmente el comportamiento que sigue es el siguiente: durante 3/4 de la conducción la temperatura del refrigerante alcanza el punto máximo, en el último cuarto baja o se estabiliza la temperatura.

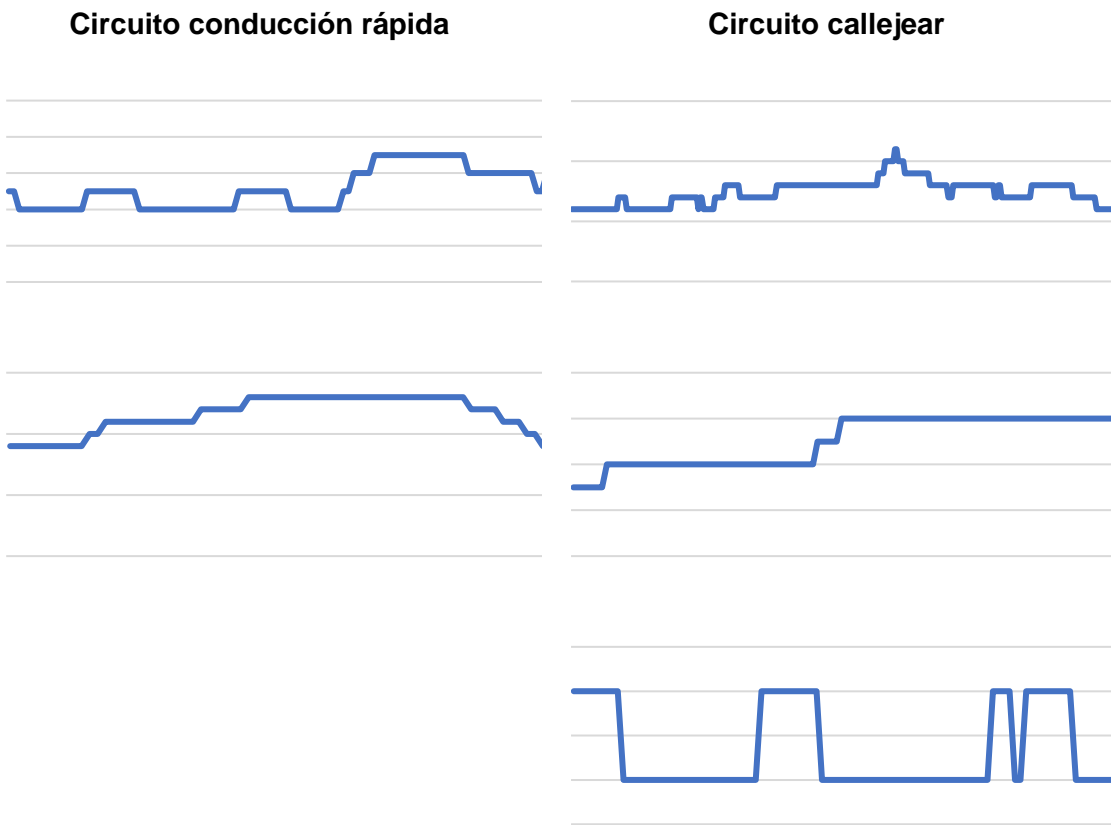


Imagen 19: Tabla comparativa temperatura del refrigerante del motor conducción rápida vs callejear

### Circuito mixto



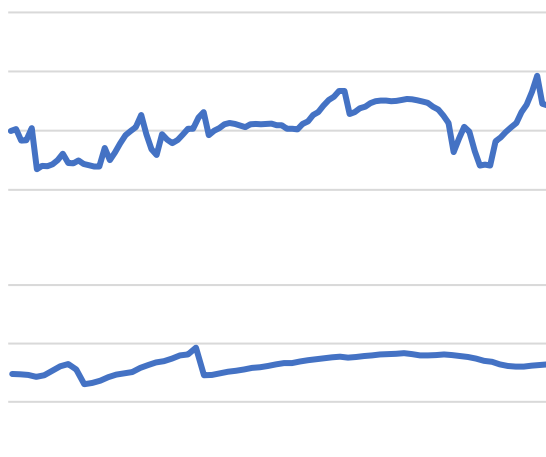
*Imagen 20: Tabla comparativa temperatura del refrigerante del motor conducción mixta*

### RPM del motor

Las revoluciones por minuto del motor es una de las variables donde la diferencia de circuitos es más clara. Podemos ver como las revoluciones por minuto del motor son mucho más estables en los circuitos de conducción rápida debido principalmente a que la velocidad es continua. En cuanto a los circuitos de callejear cabe destacar como la primera y tercera gráfica aparecen ciertos patrones de subida y bajada debido a una de las principales características de la conducción por ciudad (pasos de peatones o tráfico).

En los circuitos mixtos aparecen también estos patrones de subida y bajada además de ser un poco más estables en la primera gráfica.

### Circuito conducción rápida



### Circuito callejear

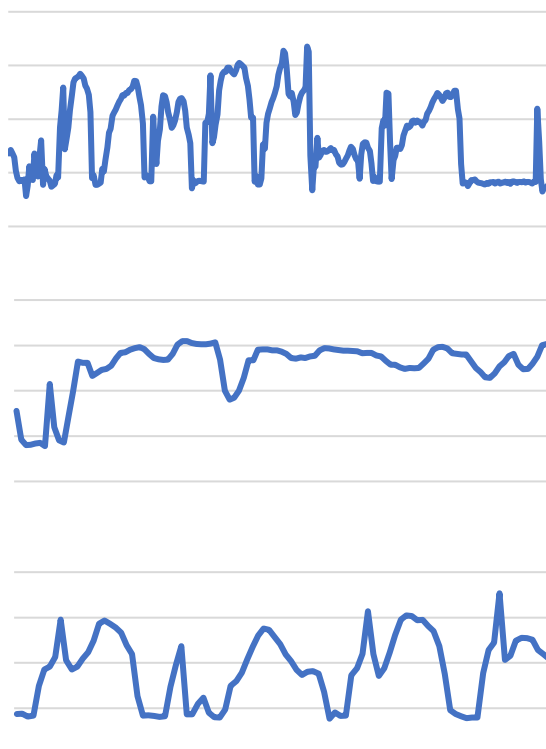


Imagen 21: Tabla comparativa RPM del motor conducción rápida vs callejear

### Circuito mixto

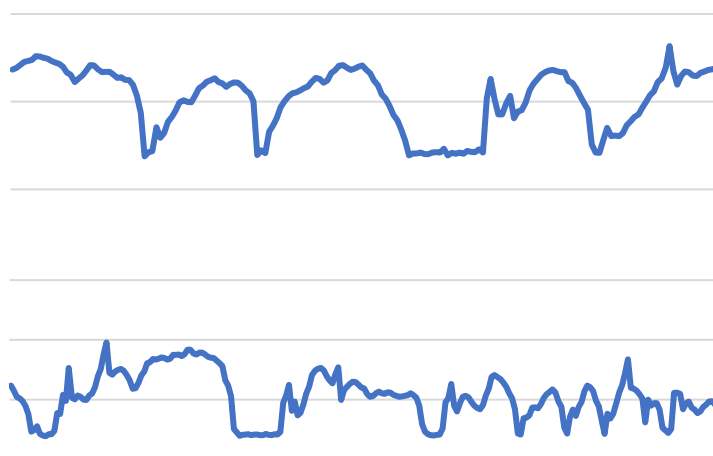


Imagen 22: Tabla comparativa RPM del motor conducción mixta

# Velocidad del vehículo

La velocidad del vehículo es otra de las variables de las cuales se pueden sacar numerosas conclusiones. En los circuitos de conducción rápida se puede apreciar como al inicio de la vía (circuito) la velocidad es menor y va aumentando a medida que se recorre el circuito, llegando a una velocidad máxima cerca del final del mismo, para descender finalmente en la salida de la vía rápida. En los circuitos de callejeo y mixtos el comportamiento es de subidas y bajadas correspondientes a situaciones de tráfico.

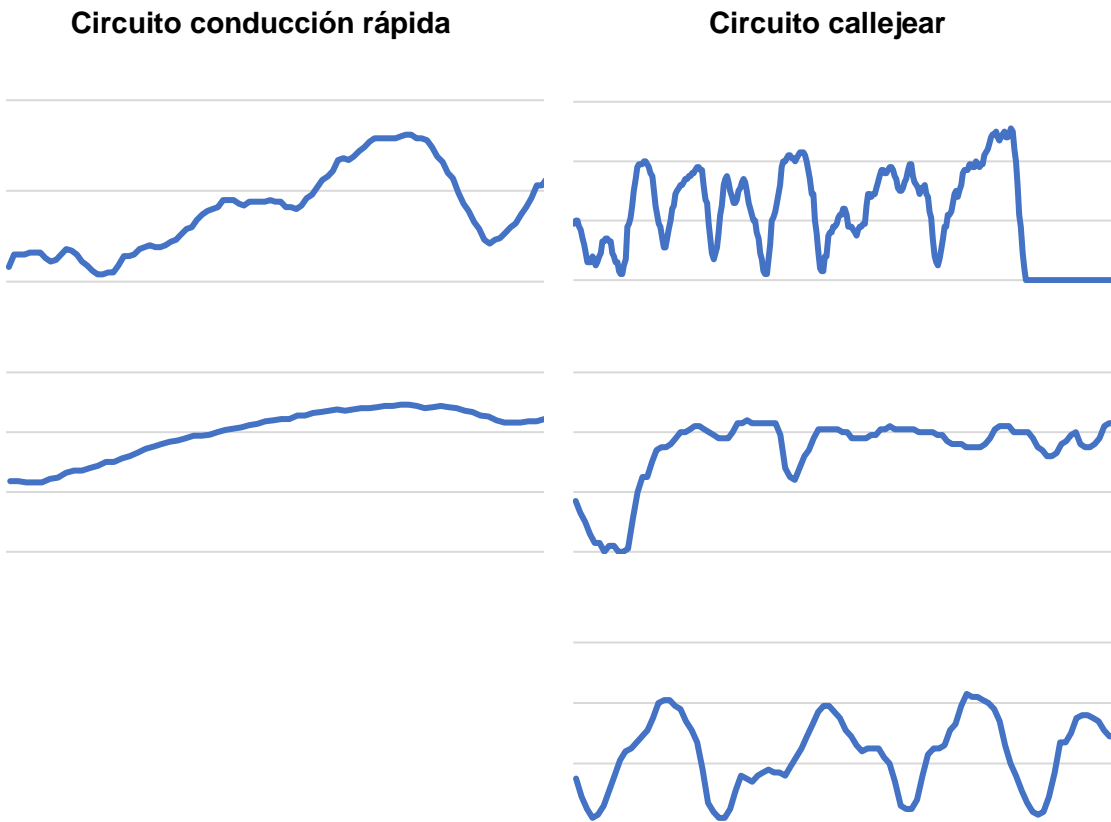
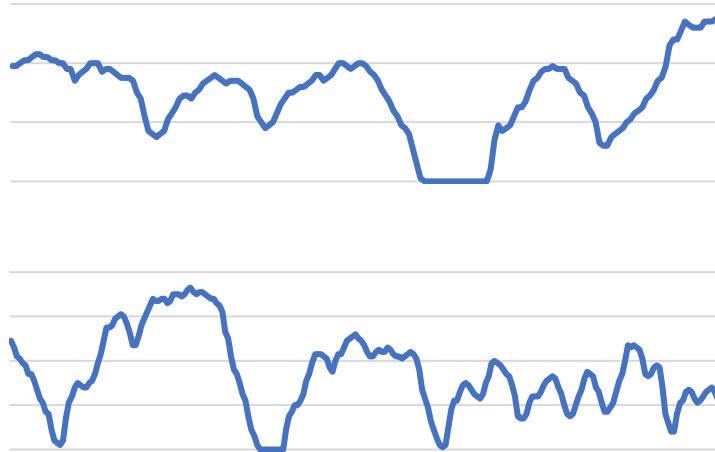


Imagen 23: Tabla comparativa velocidad del vehículo conducción rápida vs callejear

### Circuito mixto



*Imagen 24: Tabla comparativa velocidad del vehículo conducción mixta*

### Temperatura en la toma de aire

La temperatura en la toma de aire muestra ligeras diferencias entre los circuitos de conducción rápida y los circuitos de callejeo. Las diferencias no son muy notables entre ellos. Los valores en los circuitos de callejeo varían de forma más rápida que los valores en los circuitos de conducción rápida, que tienden a ser un poco más constantes.

Los circuitos mixtos tienen valores más cercanos a los circuitos de callejeo. Pero, el valor de la temperatura se mantiene más constante.

### Circuito conducción rápida



### Circuito callejear

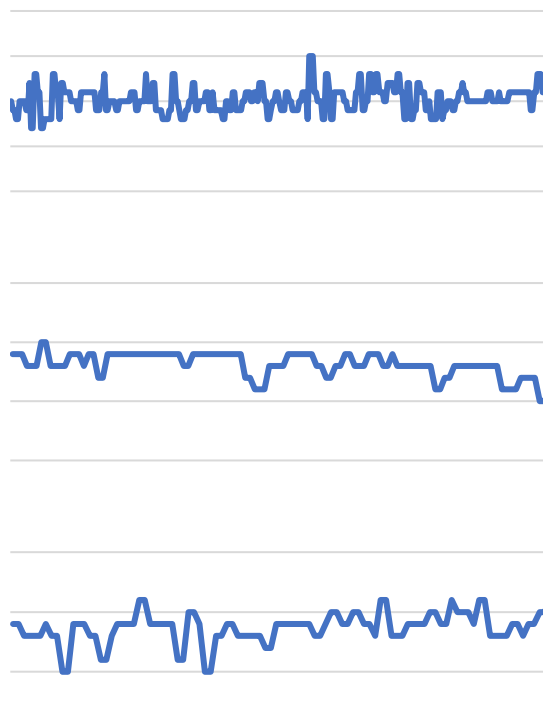


Imagen 25: Tabla comparativa temperatura en la toma de aire conducción rápida vs callejear

### Circuito mixto

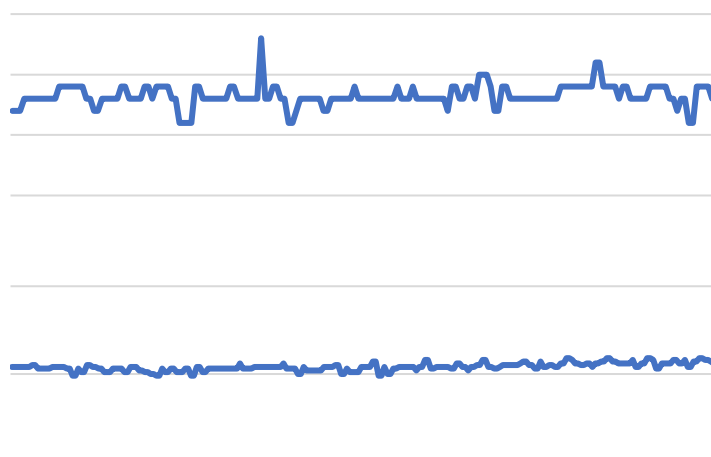


Imagen 26: Tabla comparativa temperatura en la toma de aire conducción mixta

### Posición del pedal del acelerador D

Finalmente, el valor de la posición del acelerador D, es otro de los valores que más pistas nos puede dar del circuito en el que estamos conduciendo. Comparado con los valores de rpm y velocidad del vehículo, este valor no muestra exactamente el mismo comportamiento. En los circuitos de conducción rápida el valor es más variable, para los mismos circuitos, respecto al de la velocidad del vehículo. Sin embargo, se siguen apreciando diferencias con el de callejeo. El de callejeo tiene subidas y bajadas más a menudo y con un comportamiento ligeramente más regular que los circuitos de conducción rápida.

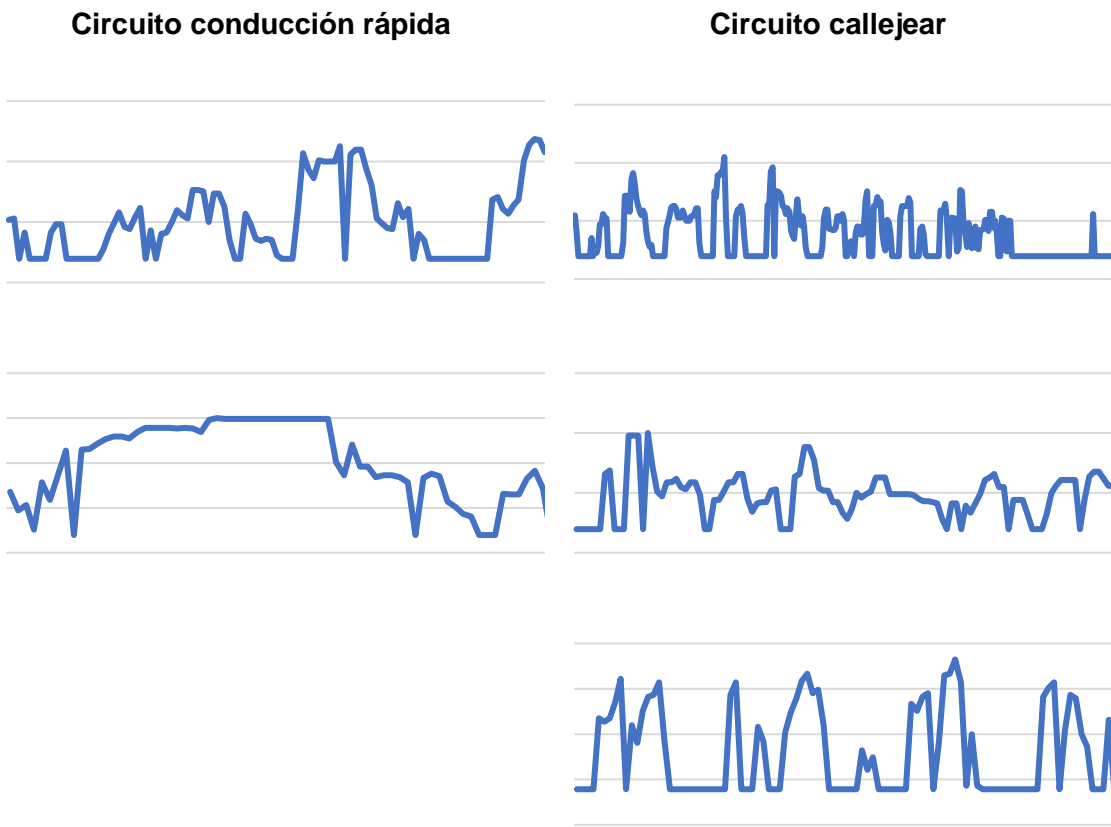
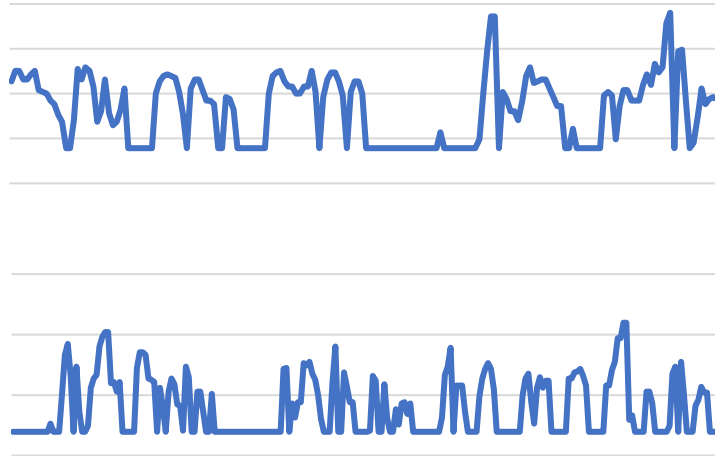


Imagen 27: Tabla comparativa posición del pedal del acelerador D conducción rápida vs callejear



### Circuito mixto



*Imagen 28: Tabla comparativa posición del pedal del acelerador D conducción mixta*

## Implementación SOM

Nuestro SOM aprenderá a representar, en un espacio de dos dimensiones, diferentes regiones del espacio de entrada donde los vectores de entrada ocurren. Tendremos un total de cien vectores de entrada. Para obtener estos cien vectores, utilizaremos los vectores de entrada formados por los valores recogidos de cada circuito. De forma que tengamos dos muestras de una conducción por vía rápida, tres muestras de una conducción de callejeo, dos muestras de un circuito mixto y 98 variaciones de las mismas.

Este paso se realizará una vez por cada variable recogida en el circuito de forma que se obtendrá finalmente seis redes SOM. Esta distinción se realiza porque de cada una de las variables se extraerán diferentes conclusiones.

Para realizar la red SOM se utilizará el programa Matlab. De esta forma se nos permite realizar un prototipo de una forma rápida. Para implementar la red Matlab dispone de herramientas especiales [41].

En primer lugar, se leerán los datos preparados del csv (nuestro dataset) y se almacenarán en una matriz.

$$M = \text{csvread}(\text{aceleracion.csv}', 0, 0);$$

Después se crea la red SOM del tamaño deseado.

$$\text{net} = \text{selforgmap}([3 \ 3]);$$

Acto seguido se entrena la red.

$$net = train(net,M);$$

Finalmente, para visualizar los resultados.

$$y = net(M);$$
$$clases = vec2ind(y);$$

La variable ‘clases’ contiene una matriz que indica a que parte del mapa forma parte una muestra. De forma que, si el SOM creado es 3 x 3, una muestra puede tomar un valor entre 4.

Además, se utilizarán las siguientes funciones de Matlab sobre el resultado de la red, con el objetivo de comprender visualmente los resultados:

- `plotsomnd`: muestra la distancia entre los vecinos. Siendo los colores oscuros estar lejos del vecino y colores claros estar cerca del vecino.
- `plotsomhits`: traza el mapa de la red con cada neurona y el número de vectores que clasifica.

## Configuración red SOM

Finalmente, para lograr un correcto funcionamiento de la red y que esta sea capaz de clasificar los datos correctamente se tiene que prestar especial atención a su configuración.

Los SOM clasifican los vectores de entrada según se agrupan en el espacio de entrada. El mapa autoorganizado aprende a reconocer estas secciones según la función de vecindad. Para ello, originalmente las posiciones de las neuronas se organizan de acuerdo con una función de topología. Esta puede ser cuadrada, hexagonal o aleatoria. Por otro lado, es necesario conocer las distancias existentes entre las neuronas. Para calcularlo, existen cuatro funciones de distancia: distancia euclídea, distancia de enlace, distancia Manhattan y “boxdist”.

Por lo tanto, existen tres parámetros a configurar para obtener una red SOM que proporcione buenos resultados: función de topología, tamaño de la red (indicado en el punto de implementación SOM) y función de distancia.

La función de topología seleccionada es del tipo red hexagonal. La red hexagonal crea un conjunto de neuronas con un patrón hexagonal. Se ha elegido este tipo de red porque existen numerosos trabajos que muestran el correcto funcionamiento de este tipo de topología que no favorece ninguna dirección en el mapa [42, 43, 44] y por su eficacia en grandes conjuntos de datos [45].

El tamaño de la red es de 3 x 3. Se ha tenido en cuenta el número de circuitos de los que se disponía añadiendo hueco para posibles situaciones de conducción ineficiente.

La función de distancia elegida es la distancia Manhattan. Se ha elegido este método porque es un método bastante probado (fiable), equilibrado y válido en las redes de tipo SOM.

## Parte 3. Resultados y clasificación

En este último apartado se mostrarán para cada uno de los parámetros de conducción los tres gráficos obtenidos resultado de la red SOM junto a una explicación del significado de dichos gráficos. Finalmente, se sacará conclusiones sobre como dichos parámetros junto a su clasificación por parte de la red permiten o no indicar si una conducción es normal o ineficiente.

### Resultados SOM carga del motor

La red SOM carga del motor es capaz de distinguir los siete circuitos junto a sus variaciones, como podemos ver en la [Imagen 29]. En la imagen de impactos, podemos ver como distingue siete diferentes apartados con 15 impactos cada uno correspondientes a cada variación del circuito.

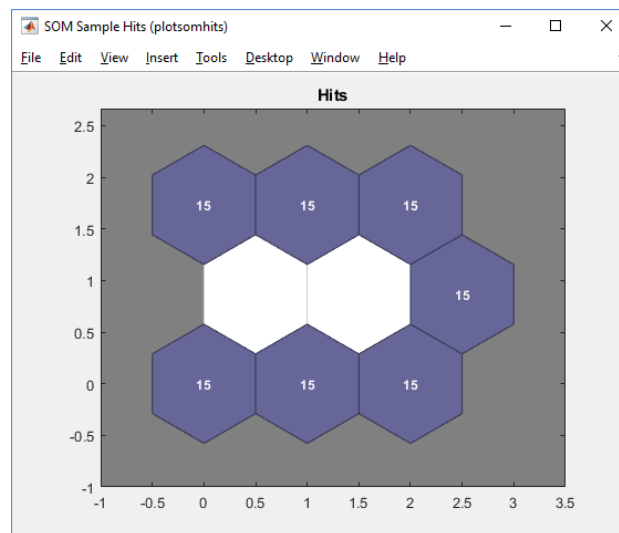
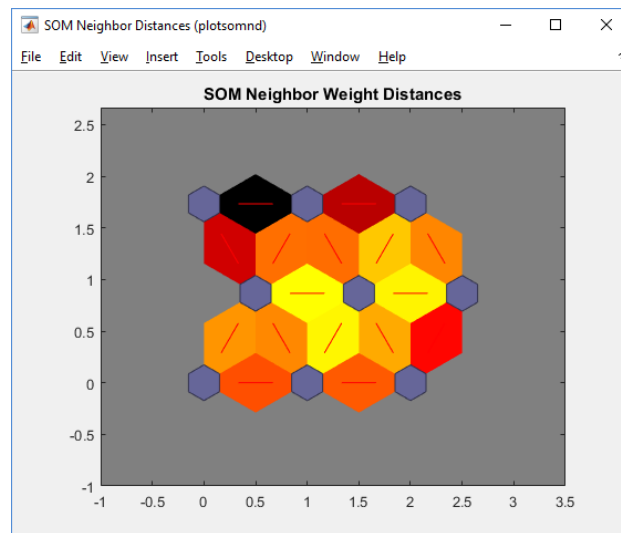


Imagen 29: SOM carga del motor, impactos

En la representación de distancias [Imagen 30] podemos ver como la red no encuentra muchas similitudes entre los diferentes circuitos puesto que sus uniones están coloreadas con colores oscuros (indicando lejanía). Comparando los resultados con la matriz de vecindad se comprueba que los circuitos más cercanos entre si son dos circuitos de callejeo (arriba-derecha). Por otro lado, los vecinos más alejados (arriba-izquierda) son un circuito rápido y un circuito mixto.



*Imagen 30: SOM carga del motor, distancias*

Por lo tanto, podemos concluir que el SOM carga del motor distingue entre los diferentes circuitos, pero, debido al gráfico de distancias y la poca capacidad de esta red de encontrar similitud entre los diferentes circuitos (puesto que la mayoría de conexiones no están próximas unas de otras), se indica que no es un valor válido para discernir entre diferentes tipos de conducciones.

## Resultados SOM temperatura del refrigerante del motor

En cuanto al refrigerante del motor, los vectores golpean en todos los lugares de la red, llenando esta y no dejando ningún hueco. Aquí la red no distingue con tanta claridad entre los distintos tipos de circuitos, esto puede ser debido a que los valores de la temperatura eran más parecidos entre si.

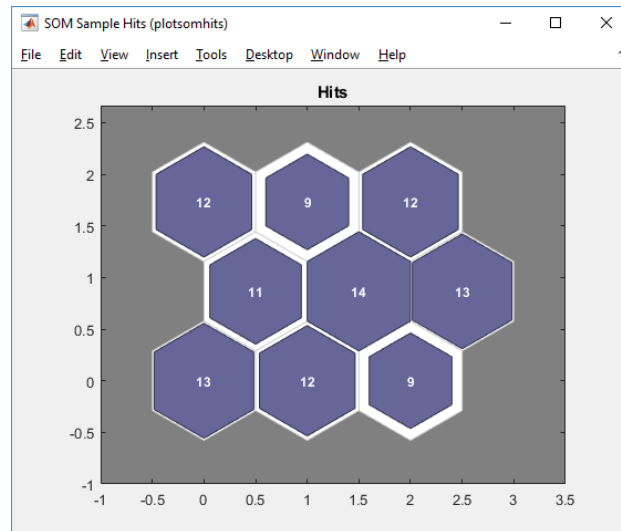


Imagen 31: SOM temperatura refrigerante del motor, impactos

En cuanto a las distancias, en este caso sí parece encontrarse una distinción entre tres grupos. La parte de la derecha donde la mayoría de vecinos están cerca entre si y la parte de la izquierda dividida entre la parte superior y la inferior.

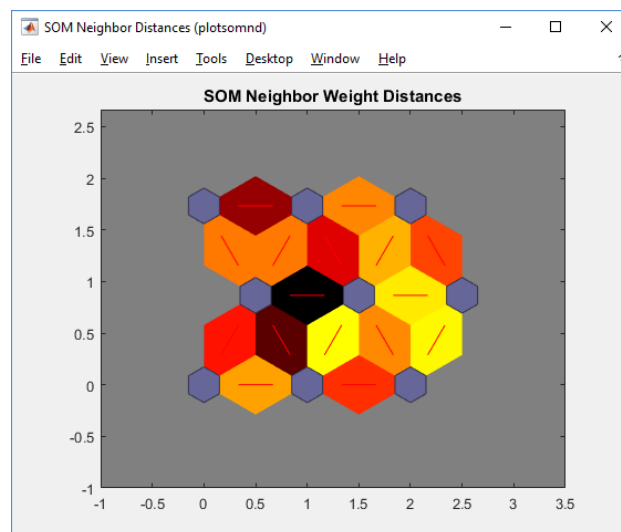


Imagen 32: SOM temperatura refrigerante del motor, distancias

Respecto a este parámetro, al parecerse tanto unos valores con otros, la red no logra dejar hueco a posibles valores anómalos del mismo. Si que existe cierta similitud mostrada en el gráfico de distancias, pero comparándolo con la matriz de vecindad no existe un patrón claro.

## Resultados SOM RPM del motor

Las RPM del motor son otra de las variables que esta red SOM consigue distinguir claramente en el gráfico de impactos. La red deja huecos donde podríamos encontrar valores referentes a conducciones ineficientes.

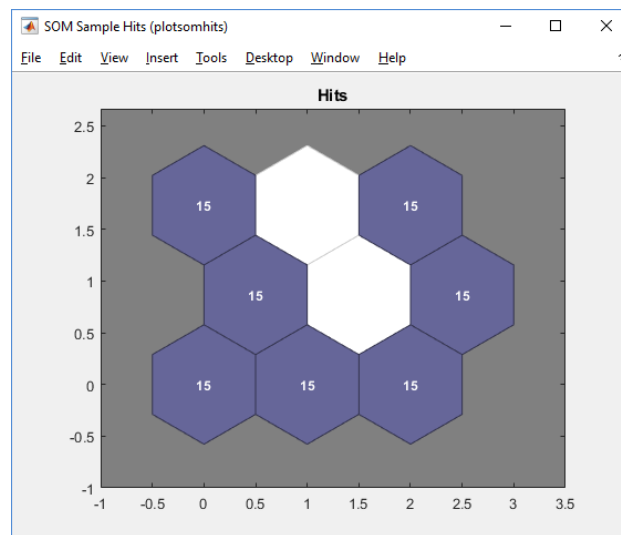


Imagen 33: SOM RPM del motor, impactos

En cuanto al gráfico de distancias, nos encontramos que los vecinos están bastante distantes entre si. Por lo que, la red logra distinguir claramente los diferentes circuitos.

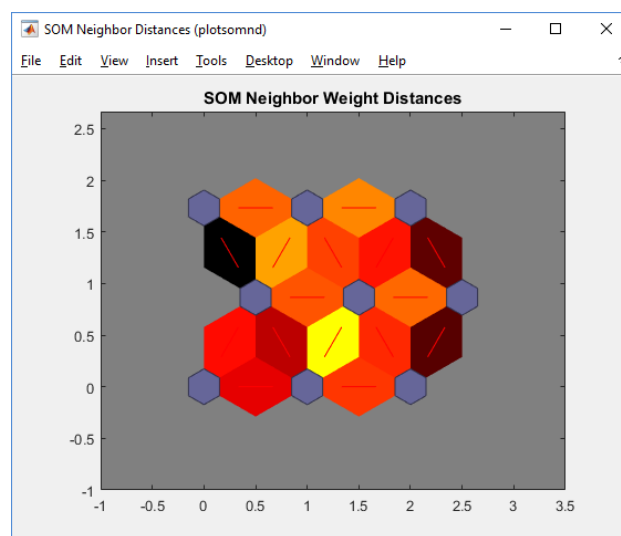
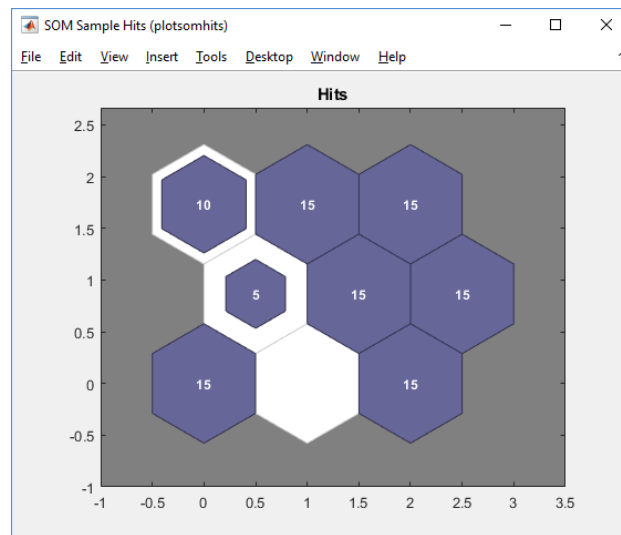


Imagen 34: SOM RPM del motor, distancias

El parámetro de RPM abre la posibilidad a su utilización para distinguir entre valores normales e ineficientes. Puesto que es capaz de distinguir claramente entre los tipos de circuitos y situaría valores anómalos lejos de estos.

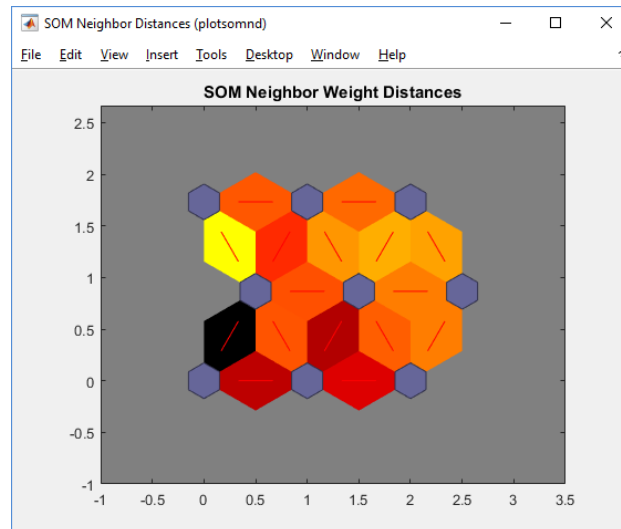
## Resultados SOM velocidad del vehículo

La velocidad del vehículo es otro de los parámetros más prometedores para la clasificación de la conducción. El gráfico de impactos [Imagen 38] nos muestra una red que distingue entre los distintos tipos de circuitos, a excepción de los circuitos mixtos que separa en dos grupos.



*Imagen 35: SOM velocidad del vehículo, impactos*

En la [Imagen 39] podemos comprobar como existe cercanía entre la mayoría de impactos situados por la parte superior y la parte derecha. Los vecinos de la parte inferior de abajo son lejanos entre ellos y el resto.



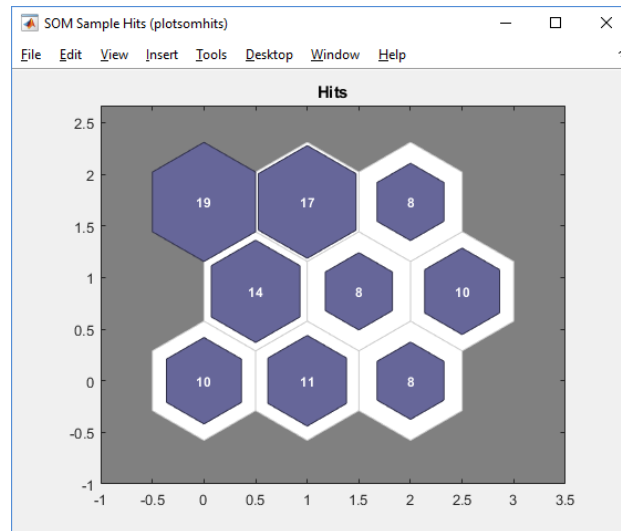
*Imagen 36: SOM velocidad del vehículo, distancias*

Combinando los resultados obtenidos por los dos gráficos podemos concluir que la velocidad del vehículo es un buen parámetro del cual indicar el estado de la conducción puesto que la mayoría de circuitos probados son cercanos entre si (encuentra similitudes o patrones en los valores de las muestras). Y deja valores sueltos en forma de hueco lejos de los demás vecinos.

## Resultados SOM temperatura en la toma de aire

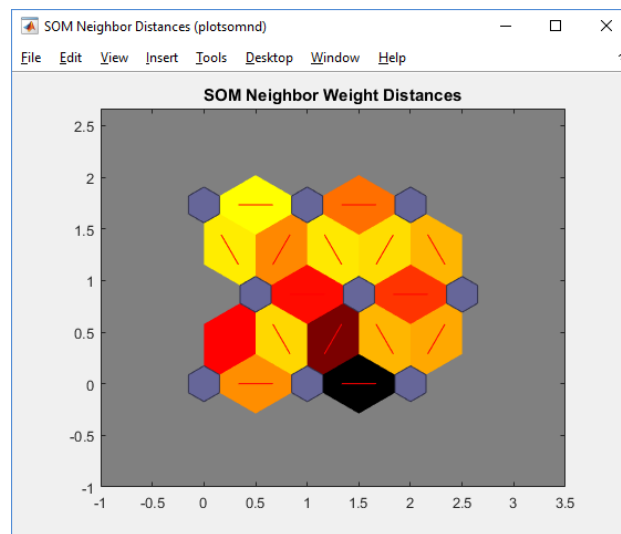
Los resultados obtenidos en el gráfico de impactos son similares a los obtenidos anteriormente en los datos de temperatura del refrigerante del motor. Estos están todos dispersos por la red sin lograr distinción entre los diferentes tipos de circuitos.





*Imagen 37: SOM temperatura toma de aire, impactos*

En cuanto al gráfico de distancias [Imagen 42] si que se encuentran similitudes en diferentes variables existiendo dos zonas en el gráfico, divididas entre la parte de arriba y la de abajo. También cabe destacar la conexión coloreado de negro indicando que es el vecino más lejano entre los impactos 11 y 8.



*Imagen 38: SOM temperatura toma de aire, distancias*

Finalmente, este valor nos proporciona más utilidad que el de temperatura del refrigerante del motor. El gráfico de distancias nos indica como al crear dos grupos claramente diferenciados pueden existir conducciones normales o ineficientes.

## Resultados SOM posición del pedal del acelerador D

Finalmente, la red SOM correspondiente a los datos de posición del pedal del acelerador D nos muestra el gráfico de impactos con mayor número de huecos entre las neuronas. Esto se debe a que agrupa dos tipos de circuitos. Comparando el gráfico con la matriz de vecindad se comprueba como los circuitos juntos corresponden a circuitos de conducción rápida y mixta.

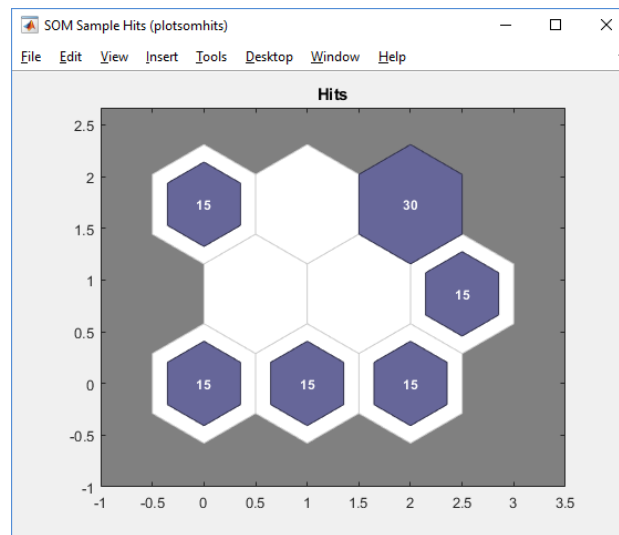


Imagen 39: SOM posición pedal acelerador D, impactos

En el gráfico de distancias podemos observar dos grupos, izquierda (vecinos 3, 5, 6, 9) y derecha (vecinos 1, 2, 4, 7, 8). También se observa vecinos muy alejados (1 con 2, 7 con 4 y 7 con 8).

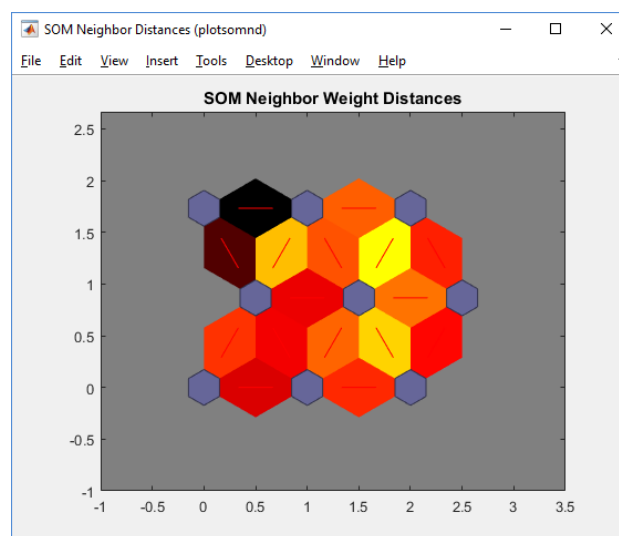


Imagen 40: SOM posición pedal acelerador D, distancias

Finalmente, la variable aceleración también parece una variable que la red SOM logra diferenciar y podría clasificar valores anómalos de la misma. Juntando los huecos dejados por los impactos y la distancia entre vecinos como el 7 con el 4 y el 8, podemos concluir que la red en este caso logra encontrar patrones entre los diferentes valores de aceleración en el pedal D.

### Clasificación

Atendiendo a los resultados obtenidos, los parámetros y las redes finales que nos permiten realizar clasificaciones y que abren la posibilidad de determinar parametros (normales e ineficientes) de conducción serían estos:

	Estilo	Combustible	Ruido	Contaminación	Estado del vehículo
Parámetros	Posición del pedal acelerador D	RPM y velocidad del vehículo	RPM	RPM	Temperatura en la toma de aire

Imagen 41: Clasificación de parámetros final

Finalmente, se quedan fuera los parámetros de carga del motor y temperatura del refrigerante del motor puesto que la red SOM no muestra patrones claros de que dichos parametros puedan servir para el proposito de la clasificación.

# Conclusiones

A partir del estudio del estado del arte y los datos obtenidos de las redes SOM, podemos concluir que es posible clasificar, a partir de parámetros obtenidos de un vehículo durante la conducción, modos de conducir.

En este trabajo, las redes SOM implementadas han sido capaces de diferenciar entre diferentes tipos de conducción, ya sea conducir por vías rápidas, callejear o realizar una conducción mezclando las anteriores. Esto abre la posibilidad de clasificar un mayor número de tipos de conducción en los que puedan entrar conducciones del tipo ineficiente.

Sin embargo, quedan abiertas numerosas cuestiones que pueden ser completadas mediante ampliaciones del trabajo en este campo:

- La escalabilidad del sistema. Queda abierta la posibilidad de escalar el sistema y probar su funcionamiento cuando el número de datos es mucho mayor. Abriendo cuestiones como: ¿Qué tipos de redes permitirían tal cantidad de datos?, ¿existe límite de vehículos de los que extraer datos?, ¿la recogida de datos en otros países o en otros modelos de vehículos sería la misma?
- Recoger más variables. En este trabajo se ha probado con 6 variables diferentes que ha proporcionado el vehículo. Sería posible probar con un mayor número de variables diferentes. Desde este punto nos podríamos preguntar: ¿Todos los vehículos recogen todas las variables?, ¿qué modelo o a partir de qué año seleccionamos los vehículos?, ¿en qué grado afecta la antigüedad de un vehículo en sus variables?
- Otras redes neuronales. En este trabajo se ha estudiado el uso de las redes SOM por sus capacidades de clasificación, pero es posible la implementación de otro tipo de redes. Probar con otro tipo de redes puede ayudar a dar perspectiva o mayor validez a la red ya tratada aquí.
- Aumentar y mejorar las muestras. Los datos recogidos en el trabajo han sido únicamente de un solo vehículo en siete trayectos diferentes. Mejorar la recolección de datos, aumentando el número de vehículos y los trayectos, añadiendo conducciones del tipo ineficiente haría que los resultados mejorasen.

# Bibliografía

- [1] M. Amarasinghe, S. Kottegoda, A. L. Arachchi, S. Muramudalige, H. M. N. D. Bandara and A. Azeez, "Cloud-based driver monitoring and vehicle diagnostic with OBD2 telematics," 2015 Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTer), Colombo, 2015, pp. 243-249.
  
- [2] Wonang Jang, Daeseong Jong and Dohoon Lee, "Methodology to improve driving habits by optimizing the in-vehicle data extracted from OBDII using genetic algorithm," 2016 Inter-national Conference on Big Data and Smart Computing (BigComp), Hong Kong, 2016, pp. 313-316.
  
- [3] W. Jang, The Self-Induction System for Fuel-Efficient Driving using OBD II. FCC 2014, Auckland, New Zealand, January 12-14, 2014.
  
- [4] M: J: Kim, J. W. Jang, and Y. S. Yu, "A study on in-vehicle system using OBD-II with navigation," Int. Journal Computer Science and Network Security, vol. 10, no. 9, Sept., 2010, pp. 136-140.
  
- [5] Berdoulat, E., Vavassori, D., and Sastre, M. Driving anger, emotional and instrumental ag-gressiveness, and impulsiveness in the prediction of aggressive and transgressive driving. Accident Analysis and Prevention 50, (2013), 758-767.
  
- [6] Aggressive driving: research update, American Automobile Assoc. Foundation for Traffic Safety, (2009).
  
- [7] Anand Paul, Naveen Chilamkurti, Alfred Daniel and Seungmin Rho, Chapter 7 – Vehicular network as business model in Big Data, In Intelligent Vehicular Networks and Communications, Elsevier, 2017, Pages 161-176, ISBN 9780128092668, <http://doi.org/10.1016/B978-0-12-809266-8.00007-7>.
  
- [8] Y. Xu and A. Helal, "Scalable Cloud–Sensor Architecture for the Internet of Things," in IEEE Internet of Things Journal, vol. 3, no. 3, pp. 285-298, June 2016.
  
- [9] T. Häberle, L. Charissis, C. Fehling, J. Nahm and F. Leymann, "The Connected Car in the Cloud: A Platform for Prototyping Telematics Services," in IEEE Software, vol. 32, no. 6, pp. 11-17, Nov.-Dec. 2015.

[10] JF Colom, H Mora, D Gil, MT Signes-Pont, Collaborative building of behavioural models based on internet of things, *Computers & Electrical Engineering* 58, 385-396, 2017

[11] K. Sasaki, N. Suzuki, S. Makido and A. Nakao, "Vehicle control system coordinated between cloud and mobile edge computing," 2016 55th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), Tsukuba, 2016, pp. 1122-1127.

[12] V Gilart-Iglesias, H Mora, R Pérez-delHoyo, C García-Mayor, A computational method based on radio frequency technologies for the analysis of accessibility of disabled people in sustainable cities, *Sustainability* 7 (11), 14935-14963, 2015.

[13] H Mora, V Gilart-Iglesias, R Pérez-del Hoyo, MD Andújar-Montoya, A Comprehensive System for Monitoring Urban Accessibility in Smart Cities, *Sensors* 17 (8), 1834, 2017.

[14] M. Gerla, E. K. Lee, G. Pau and U. Lee, "Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds," 2014 IEEE World Forum on Internet of Things (WF-IoT), Seoul, 2014, pp. 241-246.

[15] G. S. Tewolde, "Sensor and network technology for intelligent transportation systems," 2012 IEEE International Conference on Electro/Information Technology, Indianapolis, IN, 2012, pp. 1-7.

[16] H Mora, JF Colom, D Gil, A Jimeno-Morenilla, Distributed computational model for shared processing on Cyber-Physical System environments, *Computer Communications* 111, 68-83, 2017.

[17] M. Gerla, "Vehicular Cloud Computing," 2012 The 11th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), Ayia Napa, 2012, pp. 152-155.

[18] Y. T. Yu, T. Punihaole, M. Gerla and M. Y. Sanadidi, "Content routing in the Vehicle Cloud," MILCOM 2012 - 2012 IEEE Military Communications Conference, Orlando, FL, 2012, pp. 1-6.

[19] Y. Simmhan et al., "Cloud-Based Software Platform for Big Data Analytics in Smart Grids," in *Computing in Science & Engineering*, vol. 15, no. 4, pp. 38-47, July-Aug. 2013.

[20] Z. Jiao, H. Ding, M. Dang, R. Tian and B. Zhang, "Predictive Big Data Collection in Vehicular Networks: A Software Defined Networking Based Approach," 2016 IEEE Global Communications Conference (GLOBECOM), Washington, DC, 2016, pp. 1-6.

[21] H Mora-Mora, J Mora-Pascual, JM García-Chamizo, A Jimeno-Morenilla, Real-time arithmetic unit, Real-Time Systems 34 (1), 53, 2006.

[22] Shashika Ranga Muramudalige and H.M.N. Dilum Bandara. 2016. Demo: Cloud-Based Vehicular Data Analytics Platform. In Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services Companion (MobiSys '16 Companion). ACM, New York, NY, USA, 1-1

[23] Red auto-organizativa, url: [https://en.wikipedia.org/wiki/Self-organizing\\_map](https://en.wikipedia.org/wiki/Self-organizing_map)

[24] Aprendizaje competitivo, url: [https://en.wikipedia.org/wiki/Competitive\\_learning](https://en.wikipedia.org/wiki/Competitive_learning)

[25] G. G. Colombini, I. B. M. de Abreu and R. Cerri, "A self-organizing map-based method for multi-label classification," 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 2017, pp. 4291-4298.

[26] FA Pujol, H Mora, JA Girona-Selva, A connectionist computational method for face recognition, International Journal of Applied Mathematics and Computer Science 26 (2): 451-465, 2016.

[27] J. Faigl; G. A. Hollinger, "Autonomous Data Collection Using a Self-Organizing Map," in IEEE Transactions on Neural Networks and Learning Systems , vol.PP, no.99, pp.1-13.

[28] K. Okada, K. Matsui, J. Haase and H. Nishi, "Privacy-preserving data collection for demand response using self-organizing map," 2015 IEEE 13th International Conference on Industrial Informatics (INDIN), Cambridge, 2015, pp. 652-657.

[29] G. N. Iyer, S. Silas and G. Iyer, "An optimized cloud based big data processing mechanism using Self-Organizing Map in Hadoop environments," 2015 International

Conference on Green Computing and Internet of Things (ICGCIoT), Noida, 2015, pp. 244-246.

[30] R. Lasri, "Clustering and classification using a self-organizing MAP: The main flaw and the improvement perspectives," 2016 SAI Computing Conference (SAI), London, 2016, pp. 1315-1318.

[31] J Azorin-Lopez et al., Constrained self-organizing feature map to preserve feature extraction topology, Neural Computing and Applications, 1-21, 2016.

[32] D. I. Kumar and M. R. Kounte, "Comparative study of self-organizing map and deep self-organizing map using MATLAB," 2016 International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur, 2016, pp. 1020-1023.

[33] Ejemplo de sensores OBD-II, Jon N. Steiger, url: [http://www.dakota-truck.net/OBD2/obd2\\_high.html](http://www.dakota-truck.net/OBD2/obd2_high.html)

[34] OBD-II PIDs, url: [https://en.wikipedia.org/wiki/OBD-II\\_PIDs](https://en.wikipedia.org/wiki/OBD-II_PIDs)

[35] Precauciones en la normalización de datos, Santiago Morante, url: <http://www.synergicpartners.com/precauciones-a-la-hora-de-normalizar-datos-en-data-science/>

[36] Self-organizing Maps, Kevin Pang, url: <https://www.cs.hmc.edu/~kpang/nn/som.html>

[37] Self Organizing Maps 1999, Tom Germano, url: <http://davis.wpi.edu/~matt/courses/soms/>

[38] Self-Organizing Map (SOM) 1996, Jaakko Hollmen, url: <http://users.ics.aalto.fi/jhollmen/dippa/node9.html>

[39] Imagen ejemplo de un mapa autoorganizado, url: <https://es.mathworks.com/matlabcentral/fileexchange/46481-self-organizing-map-kohonen-neural-network>



[40] Pavel Stefanovič, Olga Kurasova, “Visual analysis of self-organizing maps”, 2011, *Nonlinear Analysis: Modelling and Control*, 2011, Vol. 16, No. 4, 488-504.

[41] Como implementar un mapa autoorganizado con matlab, url: <https://es.mathworks.com/help/nnet/gs/cluster-data-with-a-self-organizing-map.html>

[42] L. Middleton, J. Sivaswamy, and G. Coghill, “Logo shape discrimination using the HIP framework,” in *Proceedings of the 5th Biannual Conference on Artificial Neural Networks and Expert Systems (ANNES '01)*, pp. 59–64, 2001.

[43] Y. S. Park, J. Tison, S. Lek, J. L. Giraudel, M. Coste, and F. Delmas, “Application of a self-organizing map to select representative species in multivariate analysis: a case study determining diatom distribution patterns across France,” *Ecological Informatics*, vol. 1, no. 3, pp. 247–257, 2006.

[44] T. Kohonen, *Self-Organizing Maps*, vol. 30, Springer Series in Information Sciences, Berlin, Germany, 3rd edition, 2001, Extended Edition.

[45] A. Astel, S. Tsakovski, P. Barbieri, and V. Simeonov, “Comparison of self-organizing maps classification approach with cluster and principal components analysis for large environmental data sets,” *Water Research*, vol. 41, no. 19, pp. 4566–4578, 2007.